GDCM

2.2.0

Generated by Doxygen 1.8.1

Wed Jun 13 2012 20:40:41

Contents

Chapter 1

GDCM Documentation

This is the developpers documentation.

A PDF version of this doxygen documentation can be found here:

`http://gdcm.sourceforge.net/2.2/gdcm-2.2.0.pdf`

A tarball version of this HTML doxygen documentation can be found here:

`http://gdcm.sourceforge.net/2.2/gdcm-2.2.0-doc.tar.gz`

Author

Mathieu Malaterre

Chapter 2

gdcm2pnm

2.1 SYNOPSIS

```
gdcmdiff [options] file-in bitmap-out
```

2.2 DESCRIPTION

The **gdcm2pnm** command line program takes as input a DICOM file and produces a rendered bitmap file.

2.3 PARAMETERS

```
file-in   DICOM input filename

bitmap-out  Bitmap output filename
```

2.4 options

2.4.1 options

2.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).

  -W   --warning
         warning mode, print warning information
```

```
-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

2.5 Simple usage

gdcm2pnm will take as input DICOM and render it into a bitmap file using the window/level attributes value.

```
$ gdcm2pnm input.dcm output.png
```

It is much different from the gdcmraw or gdcmimg command line tool as it will render a DICOM image. This means that the output will be rendered in 8bits ready for display.

2.6 SEE ALSO

gdcm2vtk(1), **gdcmimg**(1)

2.7 COPYRIGHT

Chapter 3

Convert a file supported by VTK into DICOM.

3.1 SYNOPSIS

```
gdcm2vtk [options] file-in file-out
```

3.2 DESCRIPTION

The **gdcm2vtk** takes as input any file supported by VTK (including DICOM file) and will generate as output a DICOM file.

3.3 PARAMETERS

```
file-in   input filename (DICOM or VTK supported)

file-out  DICOM output filename
```

3.4 options

3.4.1 options

```
     --force-rescale    force rescale.
     --force-spacing    force spacing.
     --palette-color    when supported generate a PALETTE COLOR file.
     --argb             when supported generate a ARGB file.
     --compress         when supported generate a compressed file.
     --use-vtkdicom     Use vtkDICOMImageReader (instead of GDCM).
     --modality         set Modality.
     --lower-left       set lower left.
     --shift            set shift.
     --scale            set scale.
     --compress         set compression (MetaIO).
  -T --study-uid        Study UID.
  -S --series-uid       Series UID.
     --root-uid         Root UID.
```

3.4.2 compression options

```
-J --jpeg             Compress image in jpeg.
-K --j2k              Compress image in j2k.
-L --jpegls           Compress image in jpeg-ls.
-R --rle              Compress image in rle (lossless only).
```

3.4.3 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

3.4.4 environment variable

```
GDCM_ROOT_UID Root UID
```

3.5 DESCRIPTION

Convert a file supported by VTK into DICOM.

Typical usage is:

```
$ gdcm2vtk inputfile output.dcm
```

It uses the internal factory mechanism of VTK to recognize a file (CanRead function). See VTK supported file here:

What image file formats can VTK read and write? `http://www.vtk.org/Wiki/VTK_FAQ#What_image_-file_formats_can_VTK_read_and_write.3F`

If your input file has 4 components, the 4th comp (alpha) will be removed from the output file as DICOM does not support alpha component anymore (see –argb option).

Special care was taken for the following file format:

```
1. DICOM: Direction Cosines and vtkMedicalImageInformation are passed to the output
2. BMP: The file can be saved with a Lookup Table (see --palette-color)
3. GE Signa: vtkMedicalImageProperties is passed to the output
4. MINC: Direction Cosines is passed to the output
5. TIFF: vtkTIFFReader is currently in bad shape in VTK (different behavior in VTK 5.2 and CVS). Only use it,
```

3.5.1 CONVERT MetaImage (mhd, mha)

```
$ gdcm2vtk inputfile output.mha
```

This command will convert the input DICOM file: inputfile into a MetaImage .mha file. Same goes for .mhd file.

3.5.2 CONVERT MHA/MHD

```
$ gdcm2vtk inputfile output.mha
```

or

```
$ gdcm2vtk inputfile output.mhd
```

This command will convert the input DICOM file: inputfile into a MetaImageData .mha/.mhd file.

3.5.3 CONVERT VTI

```
$ gdcm2vtk inputfile output.vti
```

This command will convert the input DICOM file: inputfile into a XML VTK ImageData .vti file.

3.5.4 CONVERT VTK

```
$ gdcm2vtk inputfile output.vtk
```

This command will convert the input DICOM file: inputfile into an old VTK Structured PointSets .vtk file.

3.6 CONVERT DICOM

```
$ gdcm2vtk input.dcm output.dcm
```

vtkGDCMImageReader will be used to read in a DICOM file, not the default vtkDICOMImageReader. See option –use-vtkdicom to use vtkDICOMImageReader.

3.7 RoundTrip DICOM to MHD to DICOM

```
$ gdcm2vtk input_ybr.dcm output.mhd
$ gdcm2vtk --modality US --imageformat 7 output.mhd output.dcm
```

The above section shows how to convert a DICOM using the Photometric Interpretation of YBR_FULL (or even YBR_-FULL_422 is lossy) into another file format: MetaImage (mhd). Since this file format does not handle color space, we have to explicitely set it using the –imageformat command line option. The –modality command line option is required in this case since the default Secondary Capture Image Storage Class family does not allow for YBR Photometric Interpretation.

3.8 gdcm2vtk notes

IMPORTANT NOTE: The internal VTK structured will be filled from the input DICOM, and then pass to the output DICOM writer. Some information might be lost during the conversion DICOM to VTK to DICOM. This option is mostly used to test the vtkGDCMImageReader/vtkGDCMImageWriter combination.

IMPORTANT NOTE: When converting from a lossy format such as JPEG, the information of lossiness is important. The output DICOM will contains the required Lossy Image Compression attribute that indicates that image was lossy-compressed somewhere along the pipeline. See also gdcmimg (better handling of JPEG in general).

IMPORTANT NOTE: When using –use-vtkdicom the output DICOM file will always be written as MR Image Storage as this information is not available from the reader itself. This allow setting the Image Orientation (Patient) properly.

3.9 SEE ALSO

gdcmdump(1), **gdcmviewer**(1), **gdcmimg**(1)

3.10 COPYRIGHT

Chapter 4

Tool to anonymize a DICOM file.

4.1 SYNOPSIS

```
gdcmanon [options] file-in file-out
gdcmanon [options] dir-in  dir-out
```

4.2 DESCRIPTION

The **gdcmanon** tool is an implementation of PS 3.15 / E.1 / Basic Application Level Confidentiality Profile (Implementation of E.1.1 De-identify & E.1.2 Re-identify)

This tool is split into two very different operating mode:

- An implementation of PS 3.15, see -e and -d flags
- A dumb mode, see –dumb

Dumb mode and PS 3.15 do not work well together, you should really only use one type of anonymization. In case of doubt, avoid using –dumb.

In order to use the PS 3.15 implementation (-d & -e flag), you'll need a certificate to do de-identification operations, and the associated private key to do the re-identification operation. If you are only doing a one-shot anonymization and do not need to properly re-identify the DICOM file, you can safely discard the private key and only keep the certificate. See OpenSSL section below for an example on how to generate the private key/certificate pair.

gdcmanon will exit early if OpenSSL was not configured/build properly into the library (see GDCM_USE_SYSTEM_OPENSSL in cmake).

4.3 PARAMETERS

```
file-in   DICOM input filename

file-out  DICOM output filename
```

or

```
file-in   DICOM input directory

file-out  DICOM output directory
```

4.4 options

You need to specify at least one operating mode, from the following list (and only one):

4.4.1 Required parameters

```
-e --de-identify            De-identify DICOM (default)
-d --re-identify            Re-identify DICOM
   --dumb                   Dumb mode anonymizer
```

Warning when operating in dumb mode, you need to also specify an operation to do, such as 'remove' or 'empty' a tag, see below the dumb mode options.

4.4.2 options

```
-i --input                  DICOM filename / directory
-o --output                 DICOM filename / directory
-r --recursive              recursively process (sub-)directories.
   --continue               Do not stop when file found is not DICOM.
   --root-uid               Root UID.
   --resources-path         Resources path.
-k --key                    Path to RSA Private Key.
-c --certificate            Path to Certificate.
```

4.4.3 encryption options

```
   --des            DES.
   --des3           Triple DES.
   --aes128         AES 128.
   --aes192         AES 192.
   --aes256         AES 256.
```

4.4.4 dumb mode options

```
   --empty   %d,%d           DICOM tag(s) to empty
   --remove  %d,%d           DICOM tag(s) to remove
   --replace %d,%d,%s        DICOM tag(s) to replace
```

4.4.5 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

4.4.6 environment variable

```
GDCM_ROOT_UID Root UID
GDCM_RESOURCES_PATH path pointing to resources files (Part3.xml, ...)
```

4.5 Typical usage

4.5.1 De-identification (anonymization, encrypt)

The only thing required for this operation is a certificate file (in PEM format).

```
$ gdcmanon --certificate certificate.pem -e original.dcm original_anonymized.dcm
```

4.5.2 Re-identification (de-anonymization,decrypt)

The only thing required for this operation is a private key (in PEM format). It is required that the private key used for the re-identification process, was the actual private key used to generate the certificate file (certificate.pem) used during the de-identification step.

```
$ gdcmanon --key privatekey.pem -d original_anonymized.dcm original_copy.dcm
```

You can then check that original.dcm and original_copy.dcm are identical.

4.5.3 Multiple files caveat

It is very important to understand the following section, when anonymizing more than one single file. When anonymizing multiple DICOM files, you are required to use the directory input. You cannot call multiple time the gdcmanon command line tool. Indeed the tool stores in memory during the process only a hash table of conversion so that each time a particular value is found it get always replaced by the same de-identified value (think: consistent Series Instance UID).

4.5.4 Dumb mode

This functionality is not described in the DICOM standard. Users are advised that improper use of that mode is not recommended, meaning that important tag can be empty/remove/replace resulting in illegal/invalid DICOM file. Only use when you know what you are doing. If you delete a Type 1 attribute, chance is that your DICOM file will be not accepted in most DICOM third party viewer. Unfortunately this is often this mode that is implemented in popular DICOM Viewer, always prefer what the DICOM standard describes, and avoid the dumb mode.

The following example shows how to use dumb mode and achieve 5 operations at the same time:

- Empty the tag (0010,0010) Patient's Name,
- Empty the tag (0010,0020) Patient ID,
- Remove the tag (0010,0040) Patient's Sex
- Remove the tag (0010,1010) Patient's Age
- Replace the tag (0010,1030) Patient's Weight with the value '10'

You are required to check which DICOM attribute is Type 1 and Type 1C, before trying to **'Empty'** or **'Remove'** a particular DICOM attribute. For the same reason, you are required to check what are valid value in a replace operation.

```
$ gdcmanon --dumb --empty 10,10 --empty 10,20 --remove 10,40 --remove 10,1010 --replace 10,1030,10 012345.002.050
```

Multiple operation of –dumb mode can take place, just reuse the output of the previous operation. Always use gdcmdump on the input and output file to check what was actually achieved. You can use a diff program to check only what changed (see **diff**(1) for example).

4.5.4.1 Irreversible Anonymization

In some very case, one would want to anonymize using the PS 3.15 mode so as to take benefit of the automatic conversion of all content that could contain Patient related information.

In the end all Patient related information has been removed and has been secretely stored in the 0400,0500 DICOM attribute. However to make sure that no-one ever try to break that security using brute-force algorithm, one want want to remove completely this DICOM attribute. This will make the DICOM:

- Completely free of any Patient related information (as per PS 3.15 specification)
- Remove any mean of people to brute force attack the file to find out the identity of the Patient

In this case one could simply do, as a first step execute the reversible anonymizer:

```
$ gdcmanon -c certificate.pem input.dcm anonymized_reversible.dcm
```

and now completely remove the DICOM attribute containing the secretly encrypted Patient related information:

```
$ gdcmanon --dumb --remove 400,500 --remove 12,62 --remove 12,63 anonymized_reversible.dcm anonymized_irreversibl
```

4.6 OpenSSL

On most system you can have access to OpenSSL to generate the Private Key/Certificate pair.

4.6.1 Generating a Private Key

Command line to generate a rsa key (512bit)

```
$ openssl genrsa -out CA_key.pem
```

Command line to generate a rsa key (2048bit)

```
$ openssl genrsa -out CA_key.pem 2048
```

Command line to generate a rsa key (2048bit) + passphrase

```
$ openssl genrsa -des3 -out CA_key.pem 2048
```

4.6.2 Generating a Certificate

From your previously generated Private Key, you can now generate a certificate in PEM (DER format is currently not supported).

```
$ openssl req -new -key CA_key.pem -x509 -days 365 -out CA_cert.cer
```

4.7 DICOM Standard:

Page to the DICOM Standard:

`http://dicom.nema.org/`

The DICOM Standard at the time of releasing gdcmanon is:

`ftp://medical.nema.org/medical/dicom/2008/`

Direct link to PS 3.15-2008:

`ftp://medical.nema.org/medical/dicom/2008/08_15pu.pdf`

4.8 Warnings

Certains attributes may still contains Protected Health Information (PHI) after an anonymization step. This is typically the case for Patient's Address (0010,1040). The reason is that this particular attribute is not supposed to be in the composite IODs in the first place. DICOM Supp 142 includes it (however gdcmanon does not implement it).

4.9 SEE ALSO

gdcmconv(1), **gdcmdump**(1), **gdcminfo**(1), **openssl**(1)

4.10 COPYRIGHT

Chapter 5

Tool to convert DICOM to DICOM.

5.1 SYNOPSIS

```
gdcmconv [options] file-in file-out
```

5.2 DESCRIPTION

The **gdcmconv** command line program takes as input a DICOM file (file-in) and process it to generate an output DICOM file (file-out). The command line option dictate the type of operation(s) gdcmconv will use to generate the output file.

5.3 PARAMETERS

```
file-in   DICOM input filename

file-out  DICOM output filename
```

5.4 options

5.4.1 PARAMETERS

```
  -i --input      DICOM filename
  -o --output     DICOM filename
```

5.4.2 options

```
  -X --explicit            Change Transfer Syntax to explicit.
  -M --implicit            Change Transfer Syntax to implicit.
  -U --use-dict            Use dict for VR (only public by default).
     --with-private-dict   Use private dict for VR (advanced user only).
  -C --check-meta          Check File Meta Information (advanced user only).
     --root-uid            Root UID.
     --remove-gl           Remove group length (deprecated in DICOM 2008).
     --remove-private-tags Remove private tags.
     --remove-retired      Remove retired tags.
```

5.4.3 image options

```
-l --apply-lut                      Apply LUT (non-standard, advanced user only).
-P --photometric-interpretation %s  Change Photometric Interpretation (when possible).
-w --raw                            Decompress image.
-d --deflated                       Compress using deflated (gzip).
-J --jpeg                           Compress image in jpeg.
-K --j2k                            Compress image in j2k.
-L --jpegls                         Compress image in jpeg-ls.
-R --rle                            Compress image in rle (lossless only).
-F --force                          Force decompression/merging before recompression/splitting.
   --compress-icon                  Decide whether icon follows main TransferSyntax or remains uncompressed.
   --planar-configuration [01]      Change planar configuration.
-Y --lossy                          Use the lossy (if possible) compressor.
-S --split %d                       Write 2D image with multiple fragments (using max size)
```

5.4.4 JPEG options

```
-q --quality %*f           set quality.
```

5.4.5 JPEG-LS options

```
-e --lossy-error %*i       set error.
```

5.4.6 J2K options

```
-r --rate    %*f           set rate.
-q --quality %*f           set quality.
-t --tile %d,%d            set tile size.
-n --number-resolution %d  set number of resolution.
   --irreversible          set irreversible.
```

5.4.7 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

5.4.8 special options

```
-I --ignore-errors   convert even if file is corrupted (advanced users only, see disclaimers).
```

5.4.9 environment variable

```
GDCM_ROOT_UID Root UID
```

5.5 Simple usage

gdcmconv is a great tool to convert broken DICOM implementation into properly parsable DICOM file. Usage is simply:

```
$ gdcmconv input.dcm output.dcm
```

or if you prefer being explicit:

```
$ gdcmconv -i input.dcm -o output.dcm
```

Even though gdcmconv can overwrite directly on the same file (input.dcm = output.dcm), it is recommended that user should first convert into a different file to make sure the bug is properly handled by GDCM.

Typical cases where you would want to use gdcmconv in its simple form:

- convert non-cp246 conforming file into conforming cp246,
- convert implicit little endian transfer syntax file meta header into proper explicit little endian transfer syntax,
- convert the GE-13 bytes bug,
- convert dual syntax file: implicit/explicit,
- convert Philips dual Little Endian/Big Endian file,
- convert GDCM 1.2.0 broken UN-2-bytes fields,
- &...
- All other broken files listed in the supported section.

When no option other is used, only the dataset is inspected. So encapsulated Pixel Data, for instance, is not inspected for well known bugs.

When doing this kind of work, this is usually a good idea to perform some kind of quality control, see gdcmconv Quality Control section (down below).

5.6 Typical usage

5.6.1 File Meta Header

Running

```
$ gdcmconv input.dcm output.dcm
```

Is not enough to recompute file meta header, when input file is buggy. You may want to use: –check-meta

```
$ gdcmconv --check-meta input.dcm output.dcm
```

See typical cases such as: GE_DLX-8-MONO2-PrivateSyntax.dcm or PICKER-16-MONO2-No_DicomV3_Preamble.dcm from gdcmData.

5.6.2 Conversion to Explicit Transfer Syntax

To convert a file that was written using Implicit Transfer Syntax into Explicit Transfer Syntax simply use:

```
$ gdcmconv --explicit uncompressed.dcm compressed.dcm
```

5.6.3 Compressing to lossless JPEG

To compress an uncompressed DICOM file to a JPEG Lossless encapsulated format:

```
$ gdcmconv --jpeg uncompressed.dcm compressed.dcm
```

5.6.4 Compressing to lossy JPEG

To compress an uncompressed DICOM file to a JPEG Lossy encapsulated format:

```
$ gdcmconv --lossy --jpeg -q 90 uncompressed.dcm compressed.dcm
```

Note:

-q is just one of the many way to specify lossy quality, you need to inspect the other cmd line flag to specify lossyness properties.

5.6.5 Compressing to lossless JPEG-LS

To compress an uncompressed DICOM file to a JPEG-LS Lossless encapsulated format:

```
$ gdcmconv --jpegls uncompressed.dcm compressed.dcm
```

5.6.6 Compressing to lossy JPEG-LS

To compress an uncompressed DICOM file to a JPEG-LS Lossy encapsulated format:

```
$ gdcmconv --lossy --jpegls -e 2 uncompressed.dcm lossy_compressed.dcm
```

Note:

-e (or –lossy-error) means that the maximum tolerate error is 2 for each pixel value

5.6.7 Compressing to lossless J2K

To compress an uncompressed DICOM file to a JPEG-2000 Lossless encapsulated format:

```
$ gdcmconv --j2k uncompressed.dcm compressed.dcm
```

5.6.8 Compressing to lossy J2K

To compress an uncompressed DICOM file to a JPEG-2000 Lossy encapsulated format:

```
$ gdcmconv --lossy -q 55,50,45 --j2k uncompressed.dcm lossy_compressed.dcm
```

Note:

-q is just one of the many way to specify lossy quality, you need to inspect the other cmd line flag to specify lossyness properties.

5.6.9 Compressing to lossless RLE

To compress an uncompressed DICOM file to a RLE Lossless encapsulated format:

```
$ gdcmconv --rle uncompressed.dcm compressed.dcm
```

There is no such thing as lossy RLE compression.

5.6.10 Forcing (re)compression

Sometime it is necessary to use the –force option. By default when user specify –j2k and input file is already in JPEG 2000 encapsulated DICOM format then no operation takes places. By using –force you make sure that (re)compression operation takes places.

Real life example of why you would use –force:

- When Pixel Data is missing data / is padded with junk
- When you would like to make sure GDCM can handle decompression & recompression cycle

5.6.11 Decompressing a Compressed DICOM

```
$ gdcmconv --raw compressed.dcm uncompressed.dcm
```

5.6.12 Compressing an uncompressed Icon

By default when compressing a DICOM Image file, gdcmconv will not compress the icon. A user option needs to be turned on to explicitely force the compression of the Icon Image Sequence Pixel Data

For example, by default we will not compress the Icon Image Sequence Pixel Data attribute:

```
$ gdcmconv --jpeg gdcmData/simpleImageWithIcon.dcm uncompressed_icon.dcm
```

In the following example we will explicitely compress the Icon Image Sequence Pixel Data attibute. In that case the same Transfer Syntax is being used for both the main Pixel Data and the Pixel Data from the Icon Image Sequence:

```
$ gdcmconv --jpeg --compress-icon gdcmData/simpleImageWithIcon.dcm compressed_icon.dcm
```

5.6.13 Generating an Icon

For some application it might be necessary to produce a small preview of the main image to be able to quickly load that short preview instead of the main image. In that case:

```
gdcmconv --raw --generate-icon gdcmData/test.acr test_icon.dcm
```

In some cases the main Pixel Data element is expressed as pixel defined on 16bits. Since Icon can only store at most pixel of size 8bits, a rescale operation needs to take place. In order to properly select a better interval for doing the rescale operation user can specify the min max used for the rescale operation:

```
gdcmconv --raw --generate-icon --icon-minmax 0,192 gdcmData/012345.002.050.dcm icon_minmax.dcm
```

5.6.14 Changing the planar Configuration

Often RLE files are compressed using a different Planar Comnfiguration (RRR ... GGG... BBB...) instead of the usual triplet (RGB ... RGB ... RGB). So upon decompression the Planar Configuration is 1. This is perfectly legal in DICOM, however this is unconventional, and thus it may be a good idea to also change the planar configuration and set it to the default :

```
$ gdcmconv --raw --planar-configuration 0 compressed.dcm uncompressed1.dcm
```

To reinvert the planar configuration of file 'uncompressed1.dcm', simply do:

```
$ gdcmconv --raw --planar-configuration 1 uncompressed1.dcm uncompressed2.dcm
```

5.7 Lossless Conversion

When talking about lossless conversion, there is an ambiguity that need to be understood. To achieve higher compression ratio, the RGB color space is usually not used, in favor of a YBR one. Changing from one color space to the other is (bit level) not lossless.

For more detail, see what are the true lossless transformations as described:

`http://sourceforge.net/apps/mediawiki/gdcm/index.php?title=Color_Space_Transformations`

5.8 Quality Control

One important part when using gdcmconv it to have a way to quality control the output.

You can use 3rd party tool to check the output of gdcmconv is correct.

5.8.1 DCMTK / dicom3tools

Using another DICOM implementation such as the one from DCMTK or dicom3tools can be a good process to check the output of gdcmconv.

- For DCMTK use: dcmdump
- For dicom3tools use: dcdump

For reference, gdcmconv –raw will act as dcmdjpeg +cn +px, since it never tries to convert color space.

5.8.2 VIM: vimdiff

You can setup your favorite editor to compare the output, for instance in vim:

```
autocmd BufReadPre *.dcm set ro
autocmd BufReadPost *.dcm silent %!dcmdump -M +uc "%"
```

then simply do:

```
$ vimdiff input.dcm output.dcm
```

5.8.3 vbindiff

On UNIX you can visually compare binary file using the vbindiff command:

```
$ vbindiff input.dcm output.dcm
```

5.9 SEE ALSO

gdcmdump(1), **gdcmraw**(1), **gdcminfo**(1)

5.10 COPYRIGHT

Chapter 6

gdcmdiff

6.1 SYNOPSIS

```
gdcmdiff [options] file1 file2
```

6.2 DESCRIPTION

The **gdcmdiff** command line program takes as input two DICOM file: file1 and file2.

6.3 PARAMETERS

```
file1   DICOM input filename

file2  DICOM output filename
```

6.4 options

6.4.1 options

```
  -m      --meta          Compare metainformation. Default is off.
  -t <n>  --truncate <n>  String values trimmed to n characters.
```

6.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).

  -W   --warning
         warning mode, print warning information
```

```
-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

6.5 Simple usage

gdcmdiff is a great tool to diff DICOM files. Usage is simply:

```
$ gdcmdiff input1.dcm input2.dcm
```

6.6 SEE ALSO

gdcmdump(1), **gdcminfo**(1)

6.7 COPYRIGHT

Chapter 7

dumps a DICOM file, it will display the structure and values contained in the specified DICOM file.

7.1 SYNOPSIS

```
gdcmdump [options] dcm_file
gdcmdump [options] dcm_directory
```

7.2 DESCRIPTION

The **gdcmdump** command line program dumps a DICOM file to the console. For those familiar with dcmdump (DCMTK) output, gdcmdump has some minor differences. Namely:

- For Implicit Transfer Syntax gdcmdump will print ?? instead of the dictionary VR

gdcmdump has a limited private dictionary that is used to lookup private element whenever possible.

7.3 PARAMETERS

```
dcm_file        DICOM input filename
dcm_directory   DICOM input directory
```

7.4 options

7.4.1 options

```
  -x --xml-dict  generate the XML dict (only private elements for now).
  -r --recursive recursive (input is a directory)
  -d --dump      dump value (limited use).
  -p --print     print value instead of simply dumping (default).
  -c --color     print in color.
  -C --csa       print SIEMENS CSA Header (0029,[12]0,SIEMENS CSA HEADER).
  -P --pdb       print GEMS Protocol Data Block (0025,1b,GEMS_SERS_01).
     --elscint   print ELSCINT Protocol Information (01f7,26,ELSCINT1).
     --vepro     print VEPRO Protocol Information (0055,20,VEPRO VIF 3.0 DATA).
```

```
                  or VEPRO Protocol Information (0055,20,VEPRO VIM 5.0 DATA).
  -A --asn1      print encapsulated ASN1 structure >(0400,0520).
```

7.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).

  -W   --warning
         warning mode, print warning information

  -E   --error
         error mode, print error information

  -D   --debug
         debug mode, print debug information
```

7.4.3 special options

```
  -I --ignore-errors   dumps even if file is corrupted (advanced users only, see disclaimers).
```

7.5 Typical usage

7.5.1 Printing Implicit Transfer Syntax

The VR are not found in the file, thus are presented with a "(??)", and right next to it (if found) the correct VR.

Eg.:

```
$ gdcmdump GE_DLX-8-MONO2-PrivateSyntax.dcm

# Dicom-File-Format
\&...
(0008,0000) ?? (UL) 434                                          # 4,1 Generic Group Length
(0008,0005) ?? (CS) [ISO_IR 100]                                 # 10,1-n Specific Character Set
(0008,0008) ?? (CS) [ORIGINAL\\PRIMARY\\SINGLE PLANE ]             # 30,2-n Image Type
(0008,0016) ?? (UI) [1.2.840.10008.5.1.4.1.1.12.1]               # 28,1 SOP Class UID
(0008,0018) ?? (UI) [1.2.840.113619.2.16.1.0.906539207.1.24207]  # 42,1 SOP Instance UID
(0008,0020) ?? (DA) [19980923]                                   # 8,1 Study Date
(0008,0021) ?? (DA) [19980923]                                   # 8,1 Series Date
(0008,0022) ?? (DA) [19980923]                                   # 8,1 Acquisition Date
(0008,0023) ?? (DA) [19980923]                                   # 8,1 Content Date
(0008,0030) ?? (TM) [101229.000]                                 # 10,1 Study Time
(0008,0031) ?? (TM) [101229.000]                                 # 10,1 Series Time
(0008,0032) ?? (TM) [102653.000]                                 # 10,1 Acquisition Time
(0008,0033) ?? (TM) [102653.000]                                 # 10,1 Content Time
\&...
```

7.5.2 Print Private Attributes

GDCM has a limited private dictionary. Whenever possible, it will try to lookup the private data element.

```
$ gdcmdump 012345.002.050.dcm
```

```
\&...
(0009,0010) LO [GEMS_IDEN_01]                                          # 12,1 Private Creator
(0009,1001) LO [GE_GENESIS_FF ]                                        # 14,1 Full fidelity
(0009,1002) SH [MRCV]                                                  # 4,1 Suite id
(0009,1004) SH [SIGNA ]                                                # 6,1 Product id
(0009,1027) SL 985968524                                               # 4,1 Image actual date
(0009,1030) SH [19356UMR2 ]                                            # 10,1 Service id
(0009,1031) SH [999 ]                                                  # 4,1 Mobile location number
(0009,10e3) UI [1.2.840.113619.1.1.4.1762386977]                       # 32,1 Equipment UID
(0009,10e6) SH [08]                                                    # 2,1 Genesis Version - now
(0009,10e7) UL 2757786872                                              # 4,1 Exam Record checksum
(0009,10e9) SL 985968523                                               # 4,1 Actual series data time stamp
\&...
(0019,0000) UL 1208                                                    # 4,1 Generic Group Length
(0019,0010) LO [GEMS_ACQU_01]                                          # 12,1 Private Creator
(0019,100f) DS [424.399994]                                            # 10,1 Horiz. Frame of ref.
(0019,1011) SS 0                                                       # 2,1 Series contrast
\&...
(0019,10e0) DS [0.000000]                                              # 8,1 User data 24 {# DTI Diffusion Dir., relea
(0019,10e2) DS [0.000000]                                              # 8,1 Velocity Encode Scale
(0019,10f2) SS 0                                                       # 2,1 Fast phases
(0019,10f9) DS [98]                                                    # 2,1 Transmit gain
\&...
(0021,0000) UL 372                                                     # 4,1 Generic Group Length
(0021,0010) LO [GEMS_RELA_01]                                          # 12,1 Private Creator
(0021,1003) SS 0                                                       # 2,1 Series from which Prescribed
\&...
```

7.5.3 SIEMENS CSA Header

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical SIEMENS MR DICOM file.

Eg.:

```
$ gdcmdump --csa MR_SIEMENS_forceLoad29-1010_29-1020.dcm
```

```
(0029,0010)siemens csa header
Image shadow data (0029,xx10)

0 - 'EchoLinePosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '64      '
1 - 'EchoColumnPosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '64      '
2 - 'EchoPartitionPosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '32      '
3 - 'UsedChannelMask' VM 1, VR UL, SyngoDT 9, NoOfItems 6, Data '255     '
4 - 'Actual3DImaPartNumber' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
5 - 'ICE_Dims' VM 1, VR LO, SyngoDT 19, NoOfItems 6, Data 'X_1_1_1_1_1_1_31_1_1_1_1_19'
6 - 'B_value' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '0       '
7 - 'Filter1' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
8 - 'Filter2' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
\&...
```

7.5.4 GEMS Protocol Data Block

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical GEMS MR DICOM file.

Protocol Data Block : 0025,xx1b,GEMS_SERS_01

```
$ gdcmdump --pdb GE_MR_0025xx1bProtocolDataBlock.dcm
```

```
ENTRY "Head First"
POSITION "Supine"
ANREF "NA"
COIL "HEAD"
PLANE "OBLIQUE"
SEDESCFLAG "1"
SEDESC "AX FSE T2"
IMODE "2D"
PSEQ "FSE-XL"
IOPT "FC, EDR, TRF, Fast"
PLUG "22"
FILTCHOICE "None"
BWRT "-1"
TRICKSIMG "1"
TAG_SPACE "7"
TAG_TYPE "None"
\&...
```

7.5.5 ELSCINT Protocol Information

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical ELSCINT CT DICOM file.

ELSCINT Protocol Information: (01f7,26,ELSCINT1)

```
$ gdcmdump --elscint ELSCINT1_ProtocolInformation.dcm
```

```
ELSCINT1 Dumping info from tag (01f7,26,elscint1)
ELSCINT1/Item name: []
  ApprovedStep [yes]
  RefSurview [1\0]
  STD-first-img-pos [11.5]
  current-step [yes]
  ntimed-steps [0]
  orig-n-slices [390]
  protocol-file [Head_Multi_1032_usr.proc]
  protocol-name [FACE-TRAUMA/Head/Hx]
  protocol-path [/usr/diamond.root/spr/]
  protocol-step [1]
  protocol-version [2.51]

ELSCINT1/Item name: [doseright]
  ACS [n/a]
  ACS-bed-position [0]
  ACS-calc-mas [0]
  ACS-iq-parameter [0]
  ACS-learn-allowed [no]
  ACS-water-radius [-1.000000]
  ACS-water-radius-scan [-1]
\&...
```

7.5.6 VEPRO Protocol Information

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical VEPRO CT DICOM file.

ELSCINT Protocol Information: (0055,20,VEPRO VIM 5.0 DATA)

```
$ gdcmdump --vepro VEPRO_ProtocolInformation.dcm
```

```
VIMDATA2: (0055,20,VEPRO VIM 5.0 DATA)
```

```
  ID: VIM
  Version: 5.0
  UserName:
  UserAdress1: Name of Institution
  UserAdress2: Street of Institution
  UserAdress3: City of Institution
  UserAdress4:
  UserAdress5:
  RecDate: 20101001
  RecTime: 211321
  RecPlace:
  RecSource: DICOM Distributor
  DF1: P-09/10-41808
  DF2: Sultana Razia
  DF3: 19411001
  DF4: F
  DF5:
  DF6:
  DF7:
  DF8: CT Scan Brain without Contrast
  DF9: 10/10-0034873
  DF10: 10/10-00348
  DF11:
  DF12:
  DF13:
  DF14:  Head 0.5
  DF15: 4
  DF16:
  DF17:
  DF18:
  DF19:
  DF20:
  StudyUID: 1.2.392.200036.9116.2.6.1.48.1214228007.1285934880.206831
  SeriesUID: 1.2.392.200036.9116.2.6.1.48.1214228007.1285935201.938653
  Modality: CT
```

7.5.7 Philips Private MR Series Data Storage (1.3.46.670589.11.0.0.12.2)

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical Philips Private MR Series Data Storage file.

PMS Series Data Storage (2005,32,Philips MR Imaging DD 002)

```
$ gdcmdump --sds PMS_SeriesDataStorage.dcm


\&...
PMS/Item name: [PDF_CONTROL_GEN_PARS/IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_PREP_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_RECON_PARS/IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_SCAN_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_EXAM_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_HARDWARE_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_PREP_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_SPT_PARS/IEEE_PDF/Y ]
  SP_scan_resol [256\256] # 2
  SP_pda_profiles [0\0] # 2
  SP_filter [324074] # 1
  SP_analyse_with_iqt [0] # 1
```

```
  SP_main_system_type [3] # 1
  SP_gradient_system [6] # 1
  SP_coil_type [2\2\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_coil_id [2\34\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_coil_part [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_act_q [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_act_coil_freq [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_coil_m_pos [255\255\255\0\0\0\0\0\0\0\0\0\0\0\0\255] # 16
  SP_coil_t_pos [255\128\255\0\0\0\0\0\0\0\0\0\0\0\0\255] # 16
  SP_surface_coil_con [0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_proton_freq [127801349] # 1
  SP_tm_result [2\2\2\2\2\2\2\2\2\2\2\2\2\2\2\2] # 16
  SP_f0_result [0] # 1
  SP_as_result [0] # 1
  SP_po_result [0] # 1
  SP_rg_result [0] # 1
  SP_dc_result [0] # 1
  SP_ph_result [0] # 1
\&...
```

7.5.8 Encapsulated ASN1 Structure

This option is mainly used for dumping the ASN1 structure of the encrypted Attribute (0040,0520)

```
$ gdcmdump encrypted.dcm
```

```
\&...
(0400,0500) SQ                                                    # u/l,1 Encrypted Attributes Sequence
  (fffe,e000) na (Item with undefined length)
    (0400,0510) UI [1.2.840.10008.1.2]                            # 18,1 Encrypted Content Transfer Syntax UID
    (0400,0520) OB 30\\82\\03\\ba\\06\\09\\2a\\86\\48\\55\\04\\08\\13         # 958,1 Encrypted Content
  (fffe,e00d)
(fffe,e0dd)
\&...
```

```
$ gdcmdump --asn1 encrypted.dcm
```

```
    0:d=0  hl=4 l= 954 cons: SEQUENCE
    4:d=1  hl=2 l=   9 prim:  OBJECT            :pkcs7-envelopedData
   15:d=1  hl=4 l= 939 cons:  cont [ 0 ]
   19:d=2  hl=4 l= 935 cons:   SEQUENCE
   23:d=3  hl=2 l=   1 prim:    INTEGER           :00
   26:d=3  hl=4 l= 366 cons:    SET
   30:d=4  hl=4 l= 362 cons:     SEQUENCE
   34:d=5  hl=2 l=   1 prim:      INTEGER           :00
   37:d=5  hl=2 l=  82 cons:      SEQUENCE
   39:d=6  hl=2 l=  69 cons:       SEQUENCE
   41:d=7  hl=2 l=  11 cons:        SET
   43:d=8  hl=2 l=   9 cons:         SEQUENCE
   45:d=9  hl=2 l=   3 prim:          OBJECT            :countryName
   50:d=9  hl=2 l=   2 prim:          PRINTABLESTRING   :AU
   54:d=7  hl=2 l=  19 cons:        SET
   56:d=8  hl=2 l=  17 cons:         SEQUENCE
   58:d=9  hl=2 l=   3 prim:          OBJECT            :stateOrProvinceName
   63:d=9  hl=2 l=  10 prim:          PRINTABLESTRING   :Some-State
   75:d=7  hl=2 l=  33 cons:        SET
   77:d=8  hl=2 l=  31 cons:         SEQUENCE
   79:d=9  hl=2 l=   3 prim:          OBJECT            :organizationName
   84:d=9  hl=2 l=  24 prim:          PRINTABLESTRING   :Internet Widgits Pty Ltd
  110:d=6  hl=2 l=   9 prim:       INTEGER           :AC966D88787A51B4
  121:d=5  hl=2 l=  13 cons:      SEQUENCE
  123:d=6  hl=2 l=   9 prim:       OBJECT            :rsaEncryption
  134:d=6  hl=2 l=   0 prim:       NULL
  136:d=5  hl=4 l= 256 prim:      OCTET STRING      [HEX DUMP]:822368070285AD756C962ECB973514B291F946...
```

```
  396:d=3  hl=4 l= 558 cons:    SEQUENCE
  400:d=4  hl=2 l=   9 prim:     OBJECT            :pkcs7-data
  411:d=4  hl=2 l=  29 cons:     SEQUENCE
  413:d=5  hl=2 l=   9 prim:      OBJECT            :aes-256-cbc
  424:d=5  hl=2 l=  16 prim:      OCTET STRING      [HEX DUMP]:3B49AFE71749F2BFF1519EBAEA95A393
  442:d=4  hl=4 l= 512 prim:     cont [ 0 ]
```

7.6 SEE ALSO

gdcmdump(1), **gdcmraw**(1), **gdcmanon**(1)

7.7 COPYRIGHT

Chapter 8

Tool to generate a DICOMDIR file from a File-Set.

8.1 SYNOPSIS

```
gdcmgendir [options] file-in file-out
```

8.2 DESCRIPTION

8.3 PARAMETERS

```
file-in   DICOM input filename

file-out  DICOM output filename
```

8.4 options

8.4.1 Parameters

8.4.2 options

```
  -i --input              DICOM filename or directory
  -o --output             DICOM filename or directory
  -r --recursive          recursive.
     --descriptor         descriptor.
     --root-uid           Root UID.
```

8.4.3 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit
```

```
-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

8.4.4 environment variable

```
GDCM_ROOT_UID Root UID
```

8.5 Typical usage

8.6 NOTE

One may have to run some preliminary steps in order to get gdcmgendir to generate the DICOMDIR file. Namely two steps:

- Batch renaming of the DICOM filename into something compatible with ISO 9660 filename convention
- Convert all DICOM file into the Explicit VR Little Endian Uncompressed (1.2.840.10008.1.2.1)

Step 1. can be solved in a numerous way. Eg. on UNIX environment this could either be solved using the mkisofs command line tool. filename should not contains any extension since the VR CS does not allow for the '.' character. Only upper case, digit 0-9, the space ' ' and the underscore '_' character are valid in VR CS, with a maximum of 8 bytes. Another simple tool that can be handy is 'rename' in conjunction with 'basename'.

Step 2. can simply be achieved using the gdcmconv command line tool:

```
$ for i in `ls IMG*`; do gdcmconv --raw --force $i /tmp/out/$i; done
```

8.7 SEE ALSO

gdcmconv(1), **gdcmanon**(1), **rename**(1)

8.8 COPYRIGHT

Chapter 9

Manipulate DICOM image file.

9.1 SYNOPSIS

```
gdcmimg [options] file-in file-out
```

9.2 DESCRIPTION

The **gdcmimg** command line tool can be used in two fashions:

- 1. Converting a recognized file format into its encapsulated DICOM counterpart,
- 2. Anonymizing a rectangular portion of a DICOM file.

9.3 PARAMETERS

```
file-in   input filename

file-out  output filename
```

9.4 options

9.4.1 PARAMETERS

```
  -i --input     Input filename
  -o --output    Output filename
```

9.4.2 options

```
     --endian %s       Endianness (LSB/MSB).
  -d --depth %d        Depth (8/16/32).
     --sign %s         Pixel sign (0/1).
  -s --size %d,%d      Size.
  -C --sop-class-uid   SOP Class UID (name or value).
  -T --study-uid       Study UID.
  -S --series-uid      Series UID.
     --root-uid        Root UID.
```

9.4.3 fill options

```
-R --region %d,%d    Region.
-F --fill %d         Fill with pixel value specified.
```

9.4.4 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

9.4.5 environment variable

```
GDCM_ROOT_UID Root UID
```

9.5 Supported File Format (appropriate file extension)

gdcmimg will base it's conversion process based on the file extension. Follows the list of recognized file extension. When no extension is found, DICOM file is assumed.

input format

```
* RAW       (raw, gray, rgb)
* RLE       (rle)
* PNM       (pgm, pnm, ppm)
* JPEG-LS   (jls)
* JPEG 2000 (jp2, j2k, j2c, jpc)
* JPEG      (jpg, jpeg, ljpg, ljpeg)
* DICOM     ()
```

output format:

```
* PGM       (pgm, pnm, ppm)
* DICOM     ()
```

For RAW file format, you should take special care of the –endian option. For the (old) JPEG file format, both the lossy and lossless format are supported, user should pay attention to the –sign option. For file format such as RLE or RAW, user is expected to fill in information required to find the dimension and type of input data as there is no other way to find this information. For all other file format, the properties are derived from the file format itself. PNM file are supposed to be big endian.

9.6 Typical usage

9.6.1 Remove a rectangular part of the image

To fill the region [0,100]x[0,100] of a DICOM image simply do:

```
$ gdcmimg --fill 0 --region 0,100,0,100 -i input.dcm -o output_black.dcm
```

Warning: if the Pixel Data is compressed, the image is first decompressed so that pixel can be set to 0, but it is not recompressed.

9.6.2 Convert RAW to DICOM

Recognized extension is .raw, .gray or .rgb (case insensitive)

```
$ gdcmimg --size 512,512 --depth 16 -i input.raw -o output.dcm
```

the image will be a Secondary Capture

When the input is 3 component, one need to specify explicitely the Samples Per Pixel:

```
$ gdcmimg --size 512,512 --spp 3  input_rgb.raw output_rgb.dcm
```

When the filename contains .rgb as file extension output is automatically recognized as RGB no need to specify –spp

```
$ gdcmimg --size 512,512 input.rgb output_rgb.dcm
```

You can use the **dd** cmd line to skip any header you would like to discard, for instance, if you would like to skip the first 108 bytes, simply do:

```
$ dd skip=108 bs=1 if=input.raw of=output.raw
```

9.6.3 Convert PGM/PNM/PPM to DICOM

Recognized extensions are .pgm, .pnm, .ppm (case insensitive)

```
$ gdcmimg -i input.pgm -o output.dcm
```

the image will be a Secondary Capture

9.6.4 Convert RLE to DICOM

Recognized extension is .rle (case insensitive)

```
$ gdcmimg --size 512,512 --depth 16 -i input.rle -o output.dcm
```

the image will be a Secondary Capture

9.6.5 Convert JPEG to DICOM

Recognized extensions are .jpg, .jpeg, .ljpg, .ljpeg (case insensitive)

```
$ gdcmimg -i input.ljpeg -o output.dcm
```

the image will be a Secondary Capture

9.6.6 Convert J2K to DICOM

Recognized extensions are .j2k, .jp2, .jpc (case insensitive)

```
$ gdcmimg -i input.j2k -o output.dcm
```

the image will be a Secondary Capture

9.6.7 Specifying a SOP Class UID

Instead of the default Secondary Capture Image Storage, one may want to specify, say VL Photographic Image Storage.

```
$ gdcmimg --sop-class-uid 1.2.840.10008.5.1.4.1.1.77.1.4 input.jpg output.dcm
```

9.7 Multiple Files

gdcmimg handle nicely a set of files (for instance jpeg):

```
$ gdcmimg 1.jpg 2.jpg 3.jpg 4.jpg output.dcm
```

9.8 Warning

There are a couple of issues with gdcmimg implementation: For JFIF file and JP2 file (with header) the header is copied into the Pixel Data element which is illegal for JP2. Use gdcmconv to properly re-encode a JP2/JFIF file into J2K/JPG.

```
$ gdcmimg input.jp2 output_jp2.dcm
$ gdcmconv --j2k --force output_jp2.dcm output_j2k.dcm
```

For RLE file, no check is done for crossing the row boundary. It is recommended to use gdcmconv –rle to re-encode into a proper RLE file in case of doubt.

Of course if the compression is not ok with your setup, you can always de-encapsulated the DICOM file (typically JPEG) to a non-encapsulated form, using gdcmconv:

```
$ gdcmconv --raw input_jpeg.dcm output_raw.dcm
```

9.9 SEE ALSO

gdcmdump(1), **gdcmdump**(1), **gdcmraw**(1), **convert**(1), **dd**(1)

9.10 COPYRIGHT

Chapter 10

Display meta info about the input DICOM file.

10.1 SYNOPSIS

```
gdcminfo [options] file-in
```

10.2 DESCRIPTION

The **gdcminfo** command line program takes as input a DICOM file, or a directory and process it to extract meta-information about the DICOM file processed.

10.3 PARAMETERS

```
file-in   DICOM input filename
```

10.4 options

10.4.1 options

```
  -r --recursive          recursive.
  -d --check-deflated     check if file is proper deflated syntax.
     --resources-path     Resources path.
     --md5sum             Compute md5sum of Pixel Data attribute value.
     --check-compression  check the encapsulated stream compression (lossless/lossy).
```

10.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).
```

```
  -W   --warning
         warning mode, print warning information

  -E   --error
         error mode, print error information

  -D   --debug
         debug mode, print debug information
```

10.4.3 environment variable

GDCM_RESOURCES_PATH path pointing to resources files (Part3.xml, ...)

10.5 Simple usage

10.5.1 gdcmData

Using data from gdcmData:

```
$ gdcminfo gdcmData/012345.002.050.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
NumberOfDimensions: 2
Dimensions: (256,256)
Origin: (-85,21.6,108.7)
Spacing: (0.664062,0.664062,1.5)
DirectionCosines: (1,0,0,0,0,-1)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel    :1
BitsAllocated      :16
BitsStored         :16
HighBit            :15
PixelRepresentation:1
Orientation Label: CORONAL
```

10.5.2 Davie Clunie datasets:

Using data from David Clunie datasets:

```
$ gdcminfo BRTUM001.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4.1 [Enhanced MR Image Storage]
NumberOfDimensions: 3
Dimensions: (256,256,15)
Origin: (40,-105,105)
Spacing: (0.820312,0.820312,6)
DirectionCosines: (0,1,0,0,0,-1)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel    :1
BitsAllocated      :16
BitsStored         :16
HighBit            :15
PixelRepresentation:1
Orientation Label: SAGITTAL
```

10.5.3 Checking the md5sum of the Pixel Data

After compressing a DICOM file (see gdcmconv) using a lossless compression algorithm, it is fairly easy to compare the two files for differences at DICOM attribute level. However one operation is slightly easier to do: how to make sure the compression was actually lossless ? In this case one could use the –md5sum operation.

Take an uncompressed DICOM image file:

```
$ gdcminfo --md5sum SIEMENS_ImageLocationUN.dcm
```

The tool return: 0621954acd5815e0b4f7b65fcc6506b1

Now compress this file:

```
$ gdcmconv --jpegls SIEMENS_ImageLocationUN.dcm lossless_compressed.dcm
```

and then check again the md5sum:

```
$ gdcminfo --md5sum lossless_compressed.dcm
```

The tool return: 0621954acd5815e0b4f7b65fcc6506b1

10.5.4 Checking if Pixel Data is lossless

In come environment one wish to check whether or not the DICOM file is lossless or not. It is fairly easy to do that in most cases. Only in two occasion this is not clear from the sole DICOM Attribute. When the Transfer Syntax is JPEG 2000 Image Compression (1.2.840.10008.1.2.4.91) and when the Transfer Syntax is JPEG-LS Lossy (Near-Lossless) Image Compression (1.2.840.10008.1.2.4.81).

In this case, the only solution is to open the Pixel Data element, read the specific JPEG header and check whether or not the JPEG transformation was lossless or not:

```
$ gdcminfo --check-compression gdcmData/MAROTECH_CT_JP2Lossy.dcm
```

The tool returns: "Encapsulated Stream was found to be: lossy"

10.6 SEE ALSO

gdcmdump(1), **gdcmraw**(1), **gdcmconv**(1)

10.7 COPYRIGHT

Chapter 11

Tool to convert PDF to PDF/DICOM.

11.1 SYNOPSIS

```
gdcmpdf [options] file-in file-out
```

11.2 DESCRIPTION

The **gdcmpdf** tool convert a PDF file (any PDF version) into an encapsulated PDF/DICOM file. By default it will try to read the PDF meta information stored in the PDF and convert this information to some specific DICOM fields (see below). However it may fails (eg. wrong password on encrypted PDF file) in which case empty value are used.

11.3 PARAMETERS

```
file-in   PDF input filename

file-out  DICOM output filename
```

11.4 options

11.4.1 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).

  -W   --warning
         warning mode, print warning information

  -E   --error
         error mode, print error information

  -D   --debug
```

```
        debug mode, print debug information
```

11.5 Usage Example

```
$ wget http://gdcm.sourceforge.net/gdcm.pdf
$ gdcmpdf gdcm.pdf gdcm.dcm
```

To re-extract the encapsulated pdf file:

```
 $ gdcmraw -i gdcm.dcm -t 42,11 -o gdcm.dcm.pdf
 $ diff gdcm.pdf gdcm.dcm.pdf
```

11.6 PDF Info Mapping

Here is how the PDF info is mapped to DICOM information (typical pdfinfo output):

```
Title:          GDCM Reference Manual
Subject:        Grassroots DICOM API reference
Keywords:       GDCM,DICOM,JPEG,Lossless JPEG,JPEG-LS,J2K,JPEG 2000,RLE
Author:         Mathieu Malaterre and co.
Creator:        LaTeX with hyperref package
Producer:       pdfeTeX-1.21a
CreationDate:   Tue Apr 28 15:34:26 2009
Tagged:         no
Pages:          1188
Encrypted:      no
Page size:      612 x 792 pts (letter)
File size:      13756841 bytes
Optimized:      yes
PDF version:    1.4
```

Converted to DICOM this leads to:

```
# Dicom-Data-Set
# Used TransferSyntax: Little Endian Explicit
(0008,0005) CS [ISO_IR 100]                            #  10, 1 SpecificCharacterSet
(0008,0012) DA [20090428]                              #   8, 1 InstanceCreationDate
(0008,0013) TM [182550.302631]                         #  14, 1 InstanceCreationTime
(0008,0016) UI =EncapsulatedPDFStorage                 #  30, 1 SOPClassUID
(0008,0018) UI [1.2.826.0.1.3680043.2.1143.77684293519279295928902203434919711 4] #  64, 1 SOPInstanceUID
(0008,0020) DA [20090428]                              #   8, 1 StudyDate
(0008,0023) DA [20090428]                              #   8, 1 ContentDate
(0008,002a) DT [20090428153437.000000]                 #  22, 1 AcquisitionDateTime
(0008,0030) TM [182550.302160]                         #  14, 1 StudyTime
(0008,0033) TM [153426.000000]                         #  14, 1 ContentTime
(0008,0050) SH (no value available)                    #   0, 0 AccessionNumber
(0008,0060) CS [OT]                                    #   2, 1 Modality
(0008,0064) CS [WSD]                                   #   4, 1 ConversionType
(0008,0070) LO [LaTeX with hyperref package]           #  28, 1 Manufacturer
(0008,0090) PN (no value available)                    #   0, 0 ReferringPhysiciansName
(0010,0010) PN [Mathieu Malaterre and co.]             #  26, 1 PatientsName
(0010,0020) LO (no value available)                    #   0, 0 PatientID
(0010,0030) DA (no value available)                    #   0, 0 PatientsBirthDate
(0010,0040) CS (no value available)                    #   0, 0 PatientsSex
(0018,1020) LO [pdfeTeX-1.21a]                         #  14, 1 SoftwareVersions
(0020,000d) UI [1.2.826.0.1.3680043.2.1143.186812183222341735165423248075512313 3] #  64, 1 StudyInstanceUID
(0020,000e) UI [1.2.826.0.1.3680043.2.1143.133009915082574661750784610766396431 1] #  64, 1 SeriesInstanceUID
(0020,0010) SH (no value available)                    #   0, 0 StudyID
(0020,0011) IS [1]                                     #   2, 1 SeriesNumber
```

```
(0020,0013) IS [1]                                      #   2, 1 InstanceNumber
(0028,0301) CS [YES]                                    #   4, 1 BurnedInAnnotation
(0040,a043) SQ (Sequence with explicit length #=0)      #   0, 1 ConceptNameCodeSequence
(fffe,e0dd) na (SequenceDelimitationItem for re-encod.) #   0, 0 SequenceDelimitationItem
(0042,0010) ST [GDCM Reference Manual]                  #  22, 1 DocumentTitle
(0042,0011) OB 25\\50\\44\\46\\2d\\31\\2e\\34\\0a\\25\\e7\\f3\\cf\\d3\\0a\\33\\32\\30\\37\\37\\20\\30... # 137568
(0042,0012) LO [application/pdf]                        #  16, 1 MIMETypeOfEncapsulatedDocument
```

```
$ stat gdcm.pdf
  File: 'gdcm.pdf'
  Size: 13756841        Blocks: 26912      IO Block: 4096   regular file
Device: fe01h/65025d    Inode: 2675750     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1002/mmalaterre)   Gid: ( 1002/mmalaterre)
Access: 2009-04-28 16:05:00.000000000 +0200
Modify: 2009-04-28 15:34:37.000000000 +0200
Change: 2009-04-28 16:05:00.000000000 +0200
```

Explanation for the different Date/Time mappings:

- Study Date/Time, Instance Creation Date/Time are both equal to the current time gdcmpdf tool was run,
- Acquisition Date Time is set to the Modify Time of the actual pdf file,
- Content Date/Time are set from the actual PDF header info: CreationDate.

11.7 SEE ALSO

gdcmconv(1), **gdcmraw**(1), **pdfinfo**(1)

11.8 COPYRIGHT

Chapter 12

Extract Data Element Value Field.

12.1 SYNOPSIS

```
gdcmraw [options] file-in file-out
```

12.2 DESCRIPTION

The **gdcmraw** tool is mostly used for development purpose. It is used to extract a specific binary field from a DICOM DataSet.

12.3 PARAMETERS

```
file-in   DICOM input filename

file-out  output filename
```

12.4 options

12.4.1 PARAMETERS

```
  -i --input     Input filename
  -o --output    Output filename
  -t --tag       Specify tag to extract value from.
```

12.4.2 options

```
  -S --split-frags  Split fragments into multiple files.
  -p --pattern      Specify trailing file pattern (see split-frags).
  -P --pixel-data   Pixel Data trailing 0.
```

12.4.3 general options

```
  -h   --help
```

```
        print this help text and exit

  -v   --version
        print version information and exit

  -V   --verbose
        verbose mode (warning+error).

  -W   --warning
        warning mode, print warning information

  -E   --error
        error mode, print error information

  -D   --debug
        debug mode, print debug information
```

12.5 Typical usage

12.5.1 Copy Attribute Value to file

This will extract the value at Tag (0025,101b):

```
$ gdcmraw -i GE_MR_0025xx1bProtocolDataBlock.dcm -t 25,101b -o pdb.raw
```

12.5.2 Extract Pixel Data

If you do not specify any tag, the Pixel Data element is the default one. So for instance to grab the Pixel Data from an image:

```
$ gdcmraw -i test.acr -o test.raw
```

You can then for example compute the md5sum of this pixel data (very useful):

```
$ md5sum test.raw
f845c8f283d39a0204c325654493ba53  test.raw
```

12.5.3 Encapsulated Syntax

When the Pixel Data is encapsulated, multiple fragments can be used to store a single slice image:

```
$ gdcmdump D_CLUNIE_CT1_J2KR.dcm
```

```
\&...
(7fe0,0010) OB                                                  # u/l,1 Pixel Data
  (fffe,e000) ?? []                            # 0,1 Item
  (fffe,e000) ?? ff\\4f\\ff\\51\\00\\29\\00\\00\\00\\00\\02\\00\\00\\00\\02\\00\\00\\00\\00\\00\\00\\00\\00\\00\\
  (fffe,e000) ?? 2c\\b7\\ee\\68\\de\\e3\\93\\2d\\b3\\b8\\ba\\90\\7b\\42\\3e\\f8\\42\\16\\64\\88\\46\\30\\37\\d4\\
  (fffe,e000) ?? 48\\3c\\03\\e8\\c4\\3f\\44\\e1\\8a\\5c\\73\\3b\\02\\0a\\ad\\a5\\8f\\e4\\0c\\81\\76\\a2\\d7\\1b\\
(fffe,e0dd) 0
```

In order to create a J2K image out of it, we need to extract each fragments and concatenate them:

```
$ gdcmraw -i D_CLUNIE_CT1_J2KR.dcm -o D_CLUNIE_CT1_J2KR.j2k
```

This is a valid J2K file, using the Kakadu software package:

```
$ kdu_expand -i D_CLUNIE_CT1_J2KR.j2k -o D_CLUNIE_CT1_J2KR.tiff -record D_CLUNIE_CT1_J2KR.txt
```

```
$ cat D_CLUNIE_CT1_J2KR.txt
```

```
Sprofile=PROFILE2
Scap=no
Sextensions=0
Ssize={512,512}
Sorigin={0,0}
Stiles={512,512}
Stile_origin={0,0}
Scomponents=1
Ssigned=yes
Sprecision=16
Ssampling={1,1}
Sdims={512,512}
Cycc=no
Cmct=0
Clayers=1
Cuse_sop=no
Cuse_eph=no
Corder=LRCP
Calign_blk_last={no,no}
Clevels=5
Cads=0
Cdfs=0
Cdecomp=B(-:-:-)
Creversible=yes
Ckernels=W5X3
Catk=0
Cuse_precincts=no
Cblk={64,64}
Cmodes=0
Qguard=1
Qabs_ranges=18,19,19,20,19,19,20,19,19,20,19,19,20,19,19,20

>> New attributes for tile 0:
```

12.5.4 Extract fragments as single file

Sometimes each fragments is in fact a single slice, so we would not need to concatenate them:

```
$ gdcmdump 00191113.dcm
```

```
\&...
(7fe0,0010) OB                                                    # u/l,1 Pixel Data
  (fffe,e000) ?? 00\\00\\00\\00\\6b\\38\\01\\00\\10\\77\\02\\00\\37\\b6\\03\\00\\a7\\f4\\04\\00         # 20,1 It
  (fffe,e000) ?? ff\\d8\\ff\\c3\\00\\0b\\08\\02\\00\\02\\00\\01\\00\\11\\00\\ff\\c4\\00\\1b\\00\\01\\01\\01\\01\\
  (fffe,e000) ?? ff\\d8\\ff\\c3\\00\\0b\\08\\02\\00\\02\\00\\01\\00\\11\\00\\ff\\c4\\00\\1b\\00\\01\\01\\01\\01\\
  (fffe,e000) ?? ff\\d8\\ff\\c3\\00\\0b\\08\\02\\00\\02\\00\\01\\00\\11\\00\\ff\\c4\\00\\1b\\00\\01\\01\\01\\01\\
  (fffe,e000) ?? ff\\d8\\ff\\c3\\00\\0b\\08\\02\\00\\02\\00\\01\\00\\11\\00\\ff\\c4\\00\\1b\\00\\01\\01\\01\\01\\
(fffe,e0dd) 0
```

Let’s try to extract those 4 individual Lossless jpeg individually:

```
$ gdcmraw --split-frags -i 00191113.dcm -o jpeg --pattern %02d.ljpeg
```

This will output 4 files:

```
-rw-r--r--  1 mathieu mathieu   81512 2008-08-08 22:10 jpeg03.ljpeg
-rw-r--r--  1 mathieu mathieu   81694 2008-08-08 22:10 jpeg02.ljpeg
-rw-r--r--  1 mathieu mathieu   81564 2008-08-08 22:10 jpeg01.ljpeg
-rw-r--r--  1 mathieu mathieu   79970 2008-08-08 22:10 jpeg00.ljpeg
```

12.6 Footnote about JPEG files

It is a common misunderstanding to interchange 'JPEG 8bits lossy' with simply JPEG file. The JPEG specification is much broader than simply the common lossy 8bits file (as found on internet).

You can have

- JPEG Lossy 8bits
- JPEG Lossy 12bits
- JPEG Lossless 2-16bits

Those are what is defined in ITU-T T.81, ISO/IEC IS 10918-1.

12.7 SEE ALSO

gdcmdump(1), **gdcmraw**(1)

12.8 COPYRIGHT

Chapter 13

Scan a directory containing DICOM files.

13.1 SYNOPSIS

```
gdcmscanner [options] directory
```

13.2 DESCRIPTION

The **gdcmscanner** is a command line tool to quickly extract value from a set of DICOM attribute in a DICOM File-Set.

13.2.1 PARAMETERS

```
  -d --dir       DICOM directory
  -t --tag %d,%d DICOM tag(s) to look for
```

13.2.2 options

```
  -p --print      Print output.
  -r --recursive  Recusively descend directory.
```

13.2.3 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).

  -W   --warning
         warning mode, print warning information

  -E   --error
         error mode, print error information

  -D   --debug
         debug mode, print debug information
```

13.3 Typical usage

13.4 Simple usage

In order to display all the value for Patient Name (0010,0010) in the directory name **gdcmData**, simply do:

```
$ gdcmscanner -t 10,10 -d gdcmData -p
```

13.5 Complex usage

Because gdcmscanner does not support progress, you have to wait until all files are traversed to see any results. This is quite cumbersome, on UNIX this can be worked around with the following trick:

```
$ find gdcmData -type d -exec gdcmscanner -t 10,10 -d {} -p \;
```

So all directory are locally traversed (no child directory are recursively traversed), which means results comes out much faster.

13.6 SEE ALSO

gdcmdump(1), **gdcmraw**(1)

13.7 COPYRIGHT

Chapter 14

Tool to execute a DICOM Query/Retrieve operation

14.1 SYNOPSIS

```
gdcmscu [OPTION]...[OPERATION]...HOSTNAME...[PORT]...
```

Execute a DICOM Q/R operation to HOSTNAME, using port PORT (104 when not specified)

14.2 DESCRIPTION

The **gdcmscu** command line program is the tool to execute DICOM Query/Retrieve operation. It supports:

- C-ECHO (SCU)
- C-FIND (SCU)
- C-STORE (SCU)
- C-MOVE (SCU/SCP) C-MOVE operation are executed using two different ports (one for the SCU and one for the SCP).

14.3 PARAMETERS

14.4 options

14.4.1 options

```
  -H --hostname       Hostname.
  -p --port           Port number.
     --aetitle        Set calling AE Title.
     --call           Set called AE Title.
```

14.4.2 mode options

```
     --echo           C-ECHO (default when none).
     --store          C-STORE.
```

```
   --find           C-FIND.
   --move           C-MOVE.
```

14.4.3 C-STORE options

```
-i --input          DICOM filename
-r --recursive      recursively process (sub-)directories
   --store-query    Store constructed query in file
```

14.4.4 C-FIND/C-MOVE options

```
   --patientroot    C-FIND Patient Root Model.
   --studyroot      C-FIND Study Root Model.

   --patient        C-FIND Query on Patient Info (cannot be used with --studyroot).
   --study          C-FIND Query on Study Info.
   --series         C-FIND Query on Series Info.
   --image          C-FIND Query on Image Info.
   --key            0123,4567=VALUE for specifying search criteria (wildcard allowed)
                    With --key, leave blank (ie, --key 10,10="") to retrieve values
```

14.4.5 C-MOVE options

```
-o --output     DICOM filename / directory
   --port-scp   Port for incoming associations
   --key            0123,4567=VALUE for specifying search criteria (wildcard not allowed)
                    Note that C-MOVE supports the same queries as C-FIND, but no wildcards are allowed
```

14.4.6 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
--queryhelp
       print query help
```

14.4.7 environment variable

```
GDCM_ROOT_UID Root UID
```

14.5 C-ECHO usage

gdcmscu is a great tool to test if a DICOM server is up. For example to send a C-ECHO to server dicom.example.com using port 104, use:

```
$ gdcmscu dicom.example.com
```

or if you prefer being explicit:

```
$ gdcmscu --echo dicom.example.com 104
```

Using basic security your DICOM server might require that you set the appropriate called AE-TITLE

```
$ gdcmscu --echo dicom.example.com 11112 --call SERVSCP
```

If you want to specifiy your own AE-TITLE (default is GDCMSCU), simply use:

```
$ gdcmscu --echo dicom.example.com 11112 --call SERVSCP --aetitle MYSCU
```

For example you could test on the DICOM server provided by DICOMObject team:

```
$ gdcmscu www.dicomserver.co.uk 11112
```

14.6 C-STORE usage

C-STORE is the operation that allow sending a DICOM file to a remote DICOM server. For instance to send a file called myfile.dcm

```
$ gdcmscu --store dicom.example.com 104 myfile.dcm
```

or if you prefer being explicit:

```
$ gdcmscu --store dicom.example.com 104 -i myfile.dcm
```

You can even send multiple files using the same association:

```
$ gdcmscu --store dicom.example.com 104 myfile1.dcm myfile2.dcm myfile3.dcm ...
```

14.7 C-FIND usage

gdcmscu also allow querying a DICOM server. This is the C-FIND operation, for example to find all DICOM Instance where PatientsName match a particular pattern, usage is simply:

```
$ gdcmscu --find --patient dicom.example.com 11112 --patientroot --key 10,10,"A*"
```

We also support a DCMTK compatible convention:

```
$ gdcmscu --find --patient dicom.example.com 11112 --patientroot --key 10,10="A*"
```

When an attribute is set without a value it will be part of the output result:

```
$ gdcmscu --find --patient dicom.example.com 11112 --call MI2B2 --patientroot -k 10,10="A*" -k 10,20
```

14.8 C-MOVE usage

C-MOVE is the operation to retrieve a DICOM instance from a remote DICOM server. Most of the time, it is a subsequent operation after a C-FIND query. To retrieve a DICOM instance where PatientID is ABCD1234, simply execute:

```
$ gdcmscu --move --patient --aetitle ACME1 --call ACME_STORE dicom.example.com 5678 --patientroot -k 10,20="ABCD1
```

WARNING For this operation to work you need information from the DICOM server you are communicating with. Only the DICOM server you are sending a C-MOVE query will be reponsible for sending back incoming associations (the actual C-STORE SCP). Therefore you need to make sure that you mapping of (AE-TITLE,PortNumber) is properly set on the DICOM server side as well as the port for incoming association (–port-scp).

gdcmscu does not currently support external C-STORE association (C-STORE request sent to an external SCP application).

14.9 patientroot notes

The flag –patientroot is just simply a wrapper around the syntax –key 8,52=PATIENT For instance one would write using DCMTK syntax:

```
findscu --patient dicom.example.com 11112 --key 8,52=PATIENT --key 10,10="F*"
```

This would become using GDCM syntax:

```
$ gdcmscu --find --patient dicom.example.com 11112 --patientroot --key 10,10="F*"
```

14.10 Debugging

This is sometime difficult to investigate why a connection to a remote DICOM server cannot be done. Some recommendations follows:

Always try to do a simple C-ECHO at first. If you cannot get the C-ECHO to work none of the other operations will work

Before trying to a C-MOVE operation, make sure you can execute the C-FIND equivalent query first.

When doing a C-MOVE operation you really need to communicate with the PACS admin as the C-MOVE operation is different from the other lower level operation such as HTTP/GET. When doing a C-MOVE, the server will communicate back using another chanel (could be different port) using it's internal database to map an AE-TITLE back to the destination IP. Indeed the C-MOVE operation by design does not always use your incoming IP adress to send back the resulting dataset. Instead it uses a mapping of AE-TITLE to IP adress to send back any results. So pay particular attention to the spelling of your AE-TITLE and your incoming port (which may be different from the port to connect to the server).

14.11 Port Warning

Watch out that port ranging [1-1024] are reserved for admin and not easily accessible unless granted special priviliges. Therefore the default 104 DICOM port might be accessible to all your users.

14.12 C-STORE Warnings

When constructing a C-STORE operation, gdcmscu will always use the Media Storage as found in the file to be sent. For encapsulated DICOM file (eg. RLE Lossless) the receiving SCP server might not support this compression and will legitemely refuse the C-STORE operation. In this case users have to manually convert to a non-compressed form this particular file:

```
$ gdcmconv --raw compressed.dcm non_compressed.dcm
```

14.13 C-MOVE Warnings

At the moment **gdcmscu** only supports non-compressed transfer syntax. It will always request DataSet using Implicit VR Little Endian Transfer Syntax during a C-MOVE operation.

14.14 C-FIND IMAGE level (Composite Object Instance)

One should pay attention that gdcmscu –find and findscu are not completely equivalent. Using gdcmscu –find, all Unique Keys will be added automatically. One can therefore execute something like this:

```
$ gdcmscu --find --patientroot --image --key 8,18=1.2.3.4.5.6 dicom.example.com 11112
```

instead of the more explicit form

```
$ gdcmscu --find --patientroot --image --key 8,18=1.2.3.4.5.6 dicom.example.com 11112 --key 10,20 --key 20,d --key
```

This would also be equivalent to:

```
$ findscu --patient --key 8,52=IMAGE --key 8,18=1.2.3.4.5.6 dicom.example.com 11112 --key 10,20 --key 20,d --key
```

14.15 Storing the Query

It is also possible to store the query:

```
gdcmscu --find --patient --patientroot dicom.example.com 11112 --key 10,20="*" --key 10,10 --store-query query.dcm
```

One can then check the DataSet values send for the query:

```
$ gdcmdump query.dcm
# Dicom-File-Format

# Dicom-Meta-Information-Header
# Used TransferSyntax:

# Dicom-Data-Set
# Used TransferSyntax: 1.2.840.10008.1.2
(0008,0005) ?? (CS) [ISO_IR 192]                                  # 10,1-n Specific Character Set
(0008,0052) ?? (CS) [PATIENT ]                                    # 8,1 Query/Retrieve Level
(0010,0010) ?? (PN) (no value)                                    # 0,1 Patient's Name
(0010,0020) ?? (LO) [* ]                                          # 2,1 Patient ID
```

The Specific Character Set was set to "ISO_IR 192" as the locale encoding of the system was found automatically by gdcmscu to be UTF-8.

This means that the following command line will properly setup the Query with the appropriate Charset to be executed correctly:

```
$ gdcmscu --find --patient --patientroot dicom.example.com 11112 --key 10,10="*Jérôme*"
```

the query is always executed on the server side (SCP), some implementations does not support string matching with different Character Set.

14.16 SEE ALSO

gdcmconv(1)

14.17 COPYRIGHT

Chapter 15

Concatenate/Extract DICOM files.

15.1 SYNOPSIS

```
gdcmtar [options] file-in file-out
```

15.2 DESCRIPTION

The **gdcmtar** is a command line tool used to tar/untar multi-frames images (including SIEMENS MOSAIC file)

15.3 PARAMETERS

```
file-in   DICOM input filename

file-out  DICOM output filename
```

15.4 options

15.4.1 options

```
     --enhance    enhance (default)
  -U --unenhance  unenhance
  -M --mosaic     Split SIEMENS Mosaic image into multiple frames.
  -p --pattern    Specify trailing file pattern.
     --root-uid        Root UID.
```

15.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).
```

```
  -W   --warning
         warning mode, print warning information

  -E   --error
         error mode, print error information

  -D   --debug
         debug mode, print debug information
```

15.4.3 environment variable

```
  GDCM_ROOT_UID Root UID
```

15.5 Typical usage

15.5.1 SIEMENS Mosaic

```
$ gdcminfo MR-sonata-3D-as-Tile.dcm

MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
TransferSyntax is 1.2.840.10008.1.2.1 [Explicit VR Little Endian]
NumberOfDimensions: 2
Dimensions: (384,384,1)
\&...

$ gdcmtar --mosaic -i MR-sonata-3D-as-Tile.dcm -o mosaic --pattern %03d.dcm
```

Will output:

```
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic000.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic001.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic002.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic003.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic004.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic005.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic006.dcm
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic007.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic008.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic009.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic010.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic011.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic012.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic013.dcm
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic014.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic015.dcm
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic016.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic017.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic018.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic019.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic020.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic021.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic022.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic023.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic024.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic025.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic026.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic027.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic028.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic029.dcm
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic030.dcm
```

```
$ gdcminfo mosaic000.dcm

MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
TransferSyntax is 1.2.840.10008.1.2.1 [Explicit VR Little Endian]
NumberOfDimensions: 2
Dimensions: (64,64,1)
\&...
```

15.6 SEE ALSO

gdcmdump(1), **gdcmraw**(1), **gdcminfo**(1)

15.7 COPYRIGHT

Chapter 16

Simple DICOM viewer.

16.1 SYNOPSIS

```
gdcmviewer [options] file-in
```

16.2 DESCRIPTION

The **gdcmviewer** is a simple tool that show how to use vtkGDCMImageReader. The class that use gdcm to make a layer to VTK. **gdcmviewer** is basically only just a wrapper around VTK/GDCM.

This tool is meant for testing integration of GDCM in VTK. You should see it as a demo tool. It does compile with VTK ranging from 4.2 to 5.8, but only with VTK 5.2 (or above) can only play with the widgets (as described below).

16.3 PARAMETERS

```
file-in   DICOM input filename
```

16.4 options

16.4.1 options

```
     --force-rescale    force rescale (advanced users)
     --force-spacing    force spacing (advanced users)
  -r --recursive        Recusively descend directory
```

16.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).
```

```
-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

16.5 Typical usage

16.6 Simple usage

For now gdcmviewer should be started from a command line prompt. The next argument should be the name of the DICOM file you wish to read. For instance:

```
$ gdcmviewer -V 012345.002.050.dcm
```

gdcmviewer will try to read your file, and then print the vtk information associated with this file. Basically what kind of image you are looking at.

- ScalarType is the DICOM Real World Value type
- Dimensions is the dimension of the image
- Spacing is the spacing of the image
- NumberOfScalarComponents should be 1 for grayscale & PALETTE COLOR and 3 for RGB, YBR data.

16.7 Wiki Link

The wiki page, with color pictures can be found at: `http://sourceforge.net/apps/mediawiki/gdcm/index.-php?title=Gdcmviewer`

16.8 SEE ALSO

gdcmdump(1), **gdcm2vtk**(1)

16.9 COPYRIGHT

Chapter 17

Todo List

Class gdcm::CSAHeader

MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

Class gdcm::Overlay

Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differenciate them (other than the group tag).

Class gdcm::SequenceOfFragments

I do not enforce that Sequence of Fragments ends with a SQ end del

Class gdcm::TransferSyntax

: The implementation is completely retarded -> see gdcm::UIDs for a replacement We need: IsSupported We need preprocess of raw/xml file We need GetFullName()

Member gdcm::UIDGenerator::IsValid (const char *uid)

: Move that in DataStructureAndEncoding (see FileMetaInformation::CheckFileMetaInformation)

Chapter 18

Deprecated List

Member gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, bool inMove=false)

Member gdcm::DataElement::GetSequenceOfItems () const

Replaced by DataElement::GetValueAsSQ() as of GDCM 2.2.

Member gdcm::FileSet::AddFile (File const &)

. Does nothing

Member gdcm::TransferSyntax::GetSwapCode () const

Return the SwapCode associated with the Transfer Syntax. Be careful with the special GE private syntax the DataSet is written in little endian but the Pixel Data is in Big Endian.

Chapter 19

Bug List

Class gdcm::DICOMDIRGenerator

: There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the gdcm::Scanner does not allow us See PS 3.11 / Table D.3-2 STD-GEN Additional DICOMDIR Keys

Class gdcm::IPPSorter

There currently a couple of bug in this implementation:

Chapter 20

Namespace Index

20.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 21

Class Index

21.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 22

Class Index

22.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 23

File Index

23.1 File List

Here is a list of all files with brief descriptions:

Chapter 24

Namespace Documentation

24.1 gdcm Namespace Reference

Namespaces

- namespace network
- namespace SegmentHelper
- namespace terminal

  Class for Terminal Allow one to print in color in a shell.

Classes

- class AbortEvent
- class AnonymizeEvent

  AnonymizeEvent Special type of event triggered during the Anonymization process.
- class Anonymizer

  Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:
- class AnyEvent
- class ApplicationEntity

  ApplicationEntity.
- class ASN1

  Class for ASN1.
- class Attribute

  Attribute class This class use template metaprograming tricks to let the user know when the template instanciation does not match the public dictionary.
- class Attribute< Group, Element, TVR, VM::VM1 >
- class Attribute< Group, Element, TVR, VM::VM1_3 >
- class Attribute< Group, Element, TVR, VM::VM1_8 >
- class Attribute< Group, Element, TVR, VM::VM1_n >
- class Attribute< Group, Element, TVR, VM::VM2_2n >
- class Attribute< Group, Element, TVR, VM::VM2_n >
- class Attribute< Group, Element, TVR, VM::VM3_3n >
- class Attribute< Group, Element, TVR, VM::VM3_n >
- class AudioCodec

AudioCodec.

- class Base64

  Class for Base64.
- class BaseRootQuery
- class BasicOffsetTable

  Class to represent a BasicOffsetTable.
- class Bitmap

  Bitmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)
- class BitmapToBitmapFilter

  BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.
- class ByteBuffer

  ByteBuffer.
- class ByteSwap

  ByteSwap.
- class ByteSwapFilter

  ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??
- class ByteValue

  Class to represent binary value (array of bytes)
- class Codec

  Codec class.
- class Coder

  Coder.
- class CodeString

  CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.
- class Command

  Command superclass for callback/observer methods.
- class CommandDataSet

  Class to represent a Command DataSet.
- class CompositeNetworkFunctions

  Composite Network Functions These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:
- class ConstCharWrapper

  Do not use me.
- class CP246ExplicitDataElement

  Class to read/write a DataElement as CP246Explicit Data Element.
- class CryptographicMessageSyntax

  Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.
- class CSAElement

  Class to represent a CSA Element.
- class CSAHeader

  Class for CSAHeader.
- class CSAHeaderDict

Class to represent a map of CSAHeaderDictEntry.

- class CSAHeaderDictEntry

  Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information.
- class CSAHeaderDictException
- class Curve

  Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.
- class DataElement

  Class to represent a Data Element either Implicit or Explicit.
- class DataElementException
- class DataEvent

  DataEvent.
- class DataSet

  Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object.
- class DataSetEvent

  DataSetEvent Special type of event triggered during the DataSet store/move process.
- class DataSetHelper

  DataSetHelper (internal class, not intended for user level)
- class Decoder

  Decoder.
- class DefinedTerms

  Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.
- class Defs

  FIXME I do not like the name 'Defs'.
- class DeltaEncodingCodec

  DeltaEncodingCodec compression used by some private vendor.
- class DICOMDIR

  DICOMDIR class.
- class DICOMDIRGenerator

  DICOMDIRGenerator class This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.
- class Dict

  Class to represent a map of DictEntry.
- class DictConverter

  Class to convert a .dic file into something else:
- class DictEntry

  Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information.
- class DictPrinter

  DictPrinter class.
- class Dicts

  Class to manipulate the sum of knowledge (all the dict user load)

- class DirectionCosines

  class to handle DirectionCosines
- class Directory

  Class for manipulation directories.
- class DirectoryHelper
- class DummyValueGenerator

  Class for generating dummy value.
- class Dumper

  Codec class.
- class Element

  Element class.
- class Element< TVR, VM::VM1_2 >
- class Element< TVR, VM::VM1_n >
- class Element< TVR, VM::VM2_2n >
- class Element< TVR, VM::VM2_n >
- class Element< TVR, VM::VM3_3n >
- class Element< TVR, VM::VM3_n >
- class Element< VR::AS, VM::VM5 >
- class Element< VR::OB, VM::VM1 >
- class Element< VR::OW, VM::VM1 >
- class EncapsulatedDocument

  EncapsulatedDocument.
- class EncodingImplementation< VR::VRASCII >
- class EncodingImplementation< VR::VRBINARY >
- class EndEvent
- class EnumeratedValues

  Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:
- class Event

  superclass for callback/observer methods
- class Exception

  Exception.
- class ExitEvent
- class ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.
- class ExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element.
- class Fiducials

  Fiducials.
- class File

  a DICOM File See PS 3.10 File: A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may by written, read and/or deleted.
- class FileDerivation

  FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.
- class FileExplicitFilter

  FileExplicitFilter class After changing a file from Implicit to Explicit representation (see ImageChangeTransferSyntax) one operation is to make sure the VR of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit reprensentation, the VR is not stored directly in the file.

- class FileMetaInformation

  Class to represent a File Meta Information.
- class Filename

  Class to manipulate file name's.
- class FilenameGenerator

  FilenameGenerator.
- class FileSet

  File-set: A File-set is a collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which File IDs are unique.
- class FileWithName

  FileWithName.
- class FindPatientRootQuery

  PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.
- class FindStudyRootQuery

  FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.
- class Fragment

  Class to represent a Fragment.
- class Global

  Global.
- class GroupDict

  Class to represent the mapping from group number to its abbreviation and name.
- class IconImageFilter

  IconImageFilter This filter will extract icons from a gdcm::File This filter will loop over all known sequence (public and private) that may contains an IconImage and retrieve them. The filter will fails with a value of false if no icon can be found Since it handle both public and private icon type, one should not assume the icon is in uncompress form, some private vendor store private icon in JPEG8/JPEG12.
- class IconImageGenerator

  IconImageGenerator This filter will generate a valid Icon from the Pixel Data element (an instance of gdcm::Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:
- struct ignore_char
- class Image

  Image.
- class ImageApplyLookupTable

  ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.
- class ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.
- class ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.
- class ImageChangeTransferSyntax

  ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.
- class ImageCodec

  ImageCodec.
- class ImageConverter

  Image Converter.
- class ImageFragmentSplitter

  ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

- class ImageHelper

  ImageHelper (internal class, not intended for user level)
- class ImageReader

  ImageReader.
- class ImageToImageFilter

  ImageToImageFilter class Super class for all filter taking an image and producing an output image.
- class ImageWriter

  ImageWriter.
- class ImplicitDataElement

  Class to represent an Implicit VR Data Element.
- class InitializeEvent
- class IOD

  Class for representing a IOD.
- class IODEntry

  Class for representing a IODEntry.
- class IODs

  Class for representing a IODs.
- class IPPSorter

  IPPSorter Implement a simple Image Position (Patient) sorter, along the Image Orientation (Patient) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.
- class Item

  Class to represent an Item A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standart Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways Explicit/ Implicit.
- class IterationEvent
- class JPEG12Codec

  Class to do JPEG 12bits (lossy & lossless)
- class JPEG16Codec

  Class to do JPEG 16bits (lossless)
- class JPEG2000Codec

  Class to do JPEG 2000.
- class JPEG8Codec

  Class to do JPEG 8bits (lossy & lossless)
- class JPEGCodec

  JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: gdcm::JPEG8Codec, gdcm::JPEG12Codec & gdcm::JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case.
- class JPEGLSCodec

  JPEG-LS.
- class KAKADUCodec

  KAKADUCodec.
- class LO

  LO.
- class LookupTable

  LookupTable class.
- class Macro

  Class for representing a Macro.

- class Macros

  Class for representing a Modules.
- class MD5

  Class for MD5.
- class MediaStorage

  MediaStorage.
- class MemberCommand

  Command subclass that calls a pointer to a member function.
- class MeshPrimitive

  This class defines surface mesh primitives. It is designed from surface mesh primitives macro.
- class ModifiedEvent
- class Module

  Class for representing a Module.
- class ModuleEntry

  Class for representing a ModuleEntry.
- class Modules

  Class for representing a Modules.
- class MovePatientRootQuery

  MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.
- class MoveStudyRootQuery

  MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.
- class NestedModuleEntries

  Class for representing a NestedModuleEntries.
- class NoEvent
- class Object

  Object.
- struct OneShotReadBuf
- class Orientation

  class to handle Orientation
- class Overlay

  Overlay class.
- class ParseException

  ParseException Standard exception handling object.
- class Parser

  Parser ala XML_Parser from expat (SAX)
- class Patient

  See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.
- class PDBElement

  Class to represent a PDB Element.
- class PDBHeader

  Class for PDBHeader.
- class PDFCodec

  PDFCodec class.
- class PersonName

  PersonName class.
- class PhotometricInterpretation

  Class to represent an PhotometricInterpretation.

- class PixelFormat

  PixelFormat.
- class Pixmap

  Pixmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)
- class PixmapReader

  PixmapReader.
- class PixmapToPixmapFilter

  PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.
- class PixmapWriter

  PixmapWriter This class will takes two inputs:
- class PNMCodec

  Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/`.
- class Preamble

  DICOM Preamble (Part 10)
- class PresentationContext

  PresentationContext.
- class PresentationContextGenerator

  PresentationContextGenerator This class is responsible for generating the proper PresentationContext that will be used in subsequent operation during a DICOM Query/Retrieve association. The step of the association is very sensible as special care need to be taken to explicitly define what instance are going to be send and how they are encoded.
- class Printer

  Printer class.
- class PrivateDict

  Private Dict.
- class PrivateTag

  Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)
- class ProgressEvent

  ProgressEvent Special type of event triggered during.
- class PVRGCodec

  PVRGCodec.
- class PythonFilter

  PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.
- class QueryBase

  QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.
- class QueryFactory

  QueryFactory.h.
- class QueryImage

  QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.
- class QueryPatient

  QueryPatient contains: class to construct a patient-based query for c-find and c-move.
- class QuerySeries

  QuerySeries contains: class to construct a series-based query for c-find and c-move.
- class QueryStudy

  QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.
- class RAWCodec

RAWCodec class.

- class Reader

  Reader ala DOM (Document Object Model)

- class Rescaler

  Rescale class This class is meant to apply the linear transform of Stored Pixel Value to Real World Value. This is mostly found in CT or PET dataset, where the value are stored using one type, but need to be converted to another scale using a linear transform. There are basically two cases: In CT: the linear transform is generally integer based. E.g. the Stored Pixel Type is unsigned short 12bits, but to get Hounsfield unit, one need to apply the linear transform:

  $$RWV = 1. * SV - 1024$$

  So the best scalar to store the Real World Value will be 16 bits signed type.

- class RLECodec

  Class to do RLE.

- class Scanner

  Scanner This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

- class Segment

  This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

- class SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.

- class SegmentReader

  This class defines a segment reader. It reads attributes of group 0x0062.

- class SegmentWriter

  This class defines a segment writer. It writes attributes of group 0x0062.

- class SequenceOfFragments

  Class to represent a Sequence Of Fragments.

- class SequenceOfItems

  Class to represent a Sequence Of Items (value representation : SQ)

- class SerieHelper
- class Series

  Series.

- class ServiceClassUser

  ServiceClassUser.

- class SHA1

  Class for SHA1.

- class SimpleMemberCommand

  Command subclass that calls a pointer to a member function.

- class SimpleSubjectWatcher

  SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events.

- class SmartPointer

  Class for Smart Pointer.

- class SOPClassUIDToIOD

  Class convert a class SOP Class UID into IOD.

- class Sorter

  Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction.

- class Spacing

  Class for Spacing.

- class Spectroscopy

  Spectroscopy class.
- class SplitMosaicFilter

  SplitMosaicFilter class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA Image Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.
- class StartEvent
- struct static_assert_test
- struct STATIC_ASSERTION_FAILURE< true >
- class StreamImageReader

  StreamImageReader.
- class StreamImageWriter

  StreamImageReader.
- class String

  String.
- class StringFilter

  StringFilter StringFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.
- class Study

  Study.
- class Subject

  Subject.
- class Surface

  This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.
- class SurfaceHelper
- class SurfaceReader

  This class defines a SURFACE IE reader. It reads surface mesh module attributes.
- class SurfaceWriter

  This class defines a SURFACE IE writer. It writes surface mesh module attributes.
- class SwapCode

  SwapCode representation.
- class SwapperDoOp
- class SwapperNoOp
- class System

  Class to do system operation.
- class Table

  Table.
- class TableEntry

  TableEntry.
- class TableReader

  Class for representing a TableReader.
- class Tag

  Class to represent a DICOM Data Element (Attribute) Tag (Group, Element). Basically an uint32_t which can also be expressed as two uint16_t (group and element)
- class TagPath

  class to handle a path of tag.
- class Testing

  class for testing
- class Trace

Trace.

- class TransferSyntax

  Class to manipulate Transfer Syntax.
- class Type

  Type.
- struct UI
- class UIDGenerator

  Class for generating unique UID.
- class UIDs

  all known uids
- class UNExplicitDataElement

  Class to read/write a DataElement as UNExplicit Data Element.
- class UNExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:
- class Unpacker12Bits

  Pack/Unpack 12 bits pixel into 16bits.
- class Usage

  Usage.
- class UserEvent
- class Validate

  Validate class.
- class Value

  Class to represent the value of a Data Element.
- class ValueIO

  Class to dispatch template calls.
- class Version

  major/minor and build version
- class VL

  Value Length.
- class VM

  Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.
- class VR

  VR class This is adapted from DICOM standard The biggest difference is the INVALID VR and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict.
- class VR16ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.
- class VRVLSize$<$ 0 $>$
- class VRVLSize$<$ 1 $>$
- class Waveform

  Waveform class.
- class Writer

  Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.
- class XMLDictReader

  Class for representing a XMLDictReader.
- class XMLPrivateDictReader

  Class for representing a XMLPrivateDictReader.

Typedefs

- typedef String<'\\', 16 > AEComp
- typedef String<'\\', 64 > ASComp
- typedef bool(∗ BOOL_FUNCTION_PFILE_PFILE_POINTER)(File ∗, File ∗)
- typedef String<'\\', 16 > CSComp
- typedef String<'\\', 64 > DAComp
- typedef String<'\\', 64 > DTComp
- typedef std::vector
  < SmartPointer< FileWithName > > FileList
- typedef Bitmap IconImage
- typedef String<'\\', 64 > LOComp
- typedef String<'\\', 64 > LTComp
- typedef ModuleEntry MacroEntry
- typedef NestedModuleEntries NestedMacroEntries
- typedef String<'\\', 64 > PNComp
- typedef String<'\\', 64 > SHComp
- typedef String<'\\', 64 > STComp
- typedef String<'\\', 16 > TMComp
- typedef String<'\\', 64, 0 > UIComp
- typedef String<'\\', 64 > UTComp

Enumerations

- enum CompOperators {
  GDCM_EQUAL = 0,
  GDCM_DIFFERENT,
  GDCM_GREATER,
  GDCM_GREATEROREQUAL,
  GDCM_LESS,
  GDCM_LESSOREQUAL }
- enum ECharSet {
  eLatin1 = 0,
  eLatin2,
  eLatin3,
  eLatin4,
  eCyrillic,
  eArabic,
  eGreek,
  eHebrew,
  eLatin5,
  eJapanese,
  eThai,
  eJapaneseKanjiMultibyte,
  eJapaneseSupplementaryKanjiMultibyte,
  eKoreanHangulHanjaMultibyte,
  eUTF8,
  eGB18030 }
- enum EQueryLevel {
  ePatient,
  eStudy,
  eSeries,
  eImageOrFrame }

BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root name and date: 18 oct 2010 mmr.

- enum EQueryType {
  eFind,
  eMove }
- enum ERootType {
  ePatientRootType,
  eStudyRootType }
- enum LodModeType {
  LD_ALL = 0x00000000,
  LD_NOSEQ = 0x00000001,
  LD_NOSHADOW = 0x00000002,
  LD_NOSHADOWSEQ = 0x00000004 }

Functions

- ignore_char const backslash ('\\')
- VR::VRType GetVRFromTag (Tag const &tag)
- bool operator!= (const CodeString &ref, const CodeString &cs)
- bool operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & operator<< (std::ostream &os, const Version &v)
- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)
- std::ostream & operator<< (std::ostream &os, const SwapCode &sc)
- std::ostream & operator<< (std::ostream &os, const FileSet &f)
- std::ostream & operator<< (std::ostream &os, Event &e)

  Generic inserter operator for Event and its subclasses.
- std::ostream & operator<< (std::ostream &os, const PDBElement &val)
- std::ostream & operator<< (std::ostream &os, const CommandDataSet &val)
- std::ostream & operator<< (std::ostream &os, const Orientation &o)
- std::ostream & operator<< (std::ostream &_os, const IODs &_val)
- std::ostream & operator<< (std::ostream &_os, const Macros &_val)
- std::ostream & operator<< (std::ostream &_os, const Modules &_val)
- std::ostream & operator<< (std::ostream &_os, const Type &val)
- std::ostream & operator<< (std::ostream &_os, const ModuleEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const GroupDict &_val)
- std::ostream & operator<< (std::ostream &_os, const IOD &_val)
- std::ostream & operator<< (std::ostream &os, const File &val)
- std::ostream & operator<< (std::ostream &_os, const Usage &val)
- std::ostream & operator<< (std::ostream &os, const Sorter &s)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDictEntry &val)
- std::ostream & operator<< (std::ostream &os, const Preamble &val)
- std::ostream & operator<< (std::ostream &os, const Dicts &d)
- std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const Macro &_val)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDict &val)
- std::ostream & operator<< (std::ostream &os, const PDBHeader &d)
- std::ostream & operator<< (std::ostream &os, const CodeString &str)
- std::ostream & operator<< (std::ostream &os, const PrivateTag &val)
- std::ostream & operator<< (std::ostream &_os, const Module &_val)
- std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &val)
- std::ostream & operator<< (std::ostream &os, const Directory &d)

- std::ostream & operator<< (std::ostream &os, const Global &g)
- std::ostream & operator<< (std::ostream &os, const Object &obj)
- std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)
- std::ostream & operator<< (std::ostream &os, const DictEntry &val)
- std::ostream & operator<< (std::ostream &os, const CSAElement &val)
- std::ostream & operator<< (std::ostream &os, const CSAHeader &d)
- std::ostream & operator<< (std::ostream &os, const VL &val)
- std::ostream & operator<< (std::ostream &_os, const TransferSyntax &ts)
- std::ostream & operator<< (std::ostream &os, const FileMetaInformation &val)
- std::ostream & operator<< (std::ostream &_os, const VM &_val)
- std::ostream & operator<< (std::ostream &os, const Fragment &val)
- std::ostream & operator<< (std::ostream &os, const Scanner &s)
- std::ostream & operator<< (std::ostream &_os, const MediaStorage &ms)
- std::ostream & operator<< (std::ostream &os, const Dict &val)
- std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)
- std::ostream & operator<< (std::ostream &_os, const VR &val)
- std::ostream & operator<< (std::ostream &_os, const UI &_val)
- std::ostream & operator<< (std::ostream &os, const DataElement &val)
- std::ostream & operator<< (std::ostream &_os, const Tag &_val)
- std::ostream & operator<< (std::ostream &os, const DataSet &val)
- std::ostream & operator<< (std::ostream &os, const Item &val)
- std::ostream & operator<< (std::ostream &os, const PrivateDict &val)
- std::ostream & operator<< (std::ostream &_os, const UIDs &uid)
- bool operator== (const CodeString &ref, const CodeString &cs)
- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
  std::istream & operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)
- std::istream & operator>> (std::istream &in, ignore_char const &ic)
- std::istream & operator>> (std::istream &_is, Tag &_val)
- template<typename Float >
  std::string to_string (Float data)
- TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN

Variables

- static Global GlobalInstance
- VRBINARY

24.1.1 Detailed Description

This header defines the classes for the AA Actions, Association Abort Related Actions (Table 9-9 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AE Actions, Association Establishment Related Actions (Table 9-6 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AR Actions, Association Release Related Actions (Table 9-8 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the DT Actions, Data Transfer Related Actions (Table 9-8 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

24.1.2 Typedef Documentation

24.1.2.1 typedef String<'\\',16> gdcm::AEComp

24.1.2.2 typedef String<'\\',64> gdcm::ASComp

24.1.2.3 typedef bool(* gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER)(File *, File *)

24.1.2.4 typedef String<'\\',16> gdcm::CSComp

24.1.2.5 typedef String<'\\',64> gdcm::DAComp

24.1.2.6 typedef String<'\\',64> gdcm::DTComp

24.1.2.7 typedef std::vector< SmartPointer<FileWithName> > gdcm::FileList

24.1.2.8 typedef Bitmap gdcm::IconImage

24.1.2.9 typedef String<'\\',64> gdcm::LOComp

24.1.2.10 typedef String<'\\',64> gdcm::LTComp

24.1.2.11 typedef ModuleEntry gdcm::MacroEntry

24.1.2.12 typedef NestedModuleEntries gdcm::NestedMacroEntries

24.1.2.13 typedef String<'\\',64> gdcm::PNComp

24.1.2.14 typedef String<'\\',64> gdcm::SHComp

24.1.2.15 typedef String<'\\',64> gdcm::STComp

24.1.2.16 typedef String<'\\',16> gdcm::TMComp

24.1.2.17 typedef String<'\\',64,0> gdcm::UIComp

24.1.2.18 typedef String<'\\',64> gdcm::UTComp

24.1.3 Enumeration Type Documentation

24.1.3.1 enum gdcm::CompOperators

Enumerator:

GDCM_EQUAL

GDCM_DIFFERENT
GDCM_GREATER
GDCM_GREATEROREQUAL
GDCM_LESS
GDCM_LESSOREQUAL

24.1.3.2 enum gdcm::ECharSet

The character sets enumerated in PS 3.3 2009 Annex C, section C.12.1.1.2 ∗The resulting character set is stored in 0008,0005 ∗The conversion to the data element is performed by the QueryFactory itself

Enumerator:

eLatin1
eLatin2
eLatin3
eLatin4
eCyrillic
eArabic
eGreek
eHebrew
eLatin5
eJapanese
eThai
eJapaneseKanjiMultibyte
eJapaneseSupplementaryKanjiMultibyte
eKoreanHangulHanjaMultibyte
eUTF8
eGB18030

24.1.3.3 enum gdcm::EQueryLevel

BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root name and date: 18 oct 2010 mmr.

This class contains the functionality used in patient c-find and c-move queries. PatientRootQuery and StudyRootQuery derive from this class.

Namely: 1) list all tags associated with a particular query type 2) produce a query dataset via tag association

Eventually, it can be used to validate a particular dataset type.

The dataset held by this object (or, really, one of its derivates) should be passed to a c-find or c-move query.

Enumerator:

ePatient
eStudy
eSeries
eImageOrFrame

24.1.3.4 enum gdcm::EQueryType

Enumerator:

eFind

eMove

24.1.3.5 enum gdcm::ERootType

Enumerator:

ePatientRootType

eStudyRootType

24.1.3.6 enum gdcm::LodModeType

Enumerator:

LD_ALL

LD_NOSEQ

LD_NOSHADOW

LD_NOSHADOWSEQ

24.1.4 Function Documentation

24.1.4.1 ignore_char const gdcm::backslash ('\\')

Referenced by gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength().

24.1.4.2 VR::VRType gdcm::GetVRFromTag (Tag const & *tag*)

24.1.4.3 bool gdcm::operator!= (const CodeString & *ref*, const CodeString & *cs*) [inline]

24.1.4.4 bool gdcm::operator!= (const DataElement & *lhs*, const DataElement & *rhs*) [inline]

24.1.4.5 std::ostream& gdcm::operator<< (std::ostream & *os*, const Version & *v*) [inline]

References gdcm::Version::Print().

24.1.4.6 std::ostream& gdcm::operator<< (std::ostream & *_os*, const NestedModuleEntries & *_val*) [inline]

References gdcm::ModuleEntry::DataElementType, gdcm::ModuleEntry::DescriptionField, and gdcm::ModuleEntry::-Name.

24.1.4.7 std::ostream& gdcm::operator<< (std::ostream & *os*, const SwapCode & *sc*) [inline]

References gdcm::SwapCode::GetSwapCodeString().

24.1.4.8 std::ostream& gdcm::operator<< (std::ostream & *os*, const FileSet & *f*) [inline]

24.1.4.9 std::ostream& gdcm::operator<< (std::ostream & *os*, Event & *e*) [inline]

Generic inserter operator for Event and its subclasses.

References gdcm::Event::Print().

24.1.4.10 std::ostream& gdcm::operator<< (std::ostream & *os*, const PDBElement & *val*) [inline]

References gdcm::PDBElement::NameField, and gdcm::PDBElement::ValueField.

24.1.4.11 std::ostream& gdcm::operator<< (std::ostream & *os*, const CommandDataSet & *val*) [inline]

References gdcm::DataSet::Print().

24.1.4.12 std::ostream& gdcm::operator<< (std::ostream & *os*, const Orientation & *o*) [inline]

References gdcm::Orientation::Print().

24.1.4.13 std::ostream& gdcm::operator<< (std::ostream & *_os*, const IODs & *_val*) [inline]

24.1.4.14 std::ostream& gdcm::operator<< (std::ostream & *_os*, const Macros & *_val*) [inline]

24.1.4.15 std::ostream& gdcm::operator<< (std::ostream & *_os*, const Modules & *_val*) [inline]

24.1.4.16 std::ostream& gdcm::operator<< (std::ostream & *_os*, const Type & *val*) [inline]

References gdcm::Type::GetTypeString().

24.1.4.17 std::ostream& gdcm::operator<< (std::ostream & *_os*, const ModuleEntry & *_val*) [inline]

References gdcm::ModuleEntry::DataElementType, gdcm::ModuleEntry::DescriptionField, and gdcm::ModuleEntry::-Name.

24.1.4.18 std::ostream& gdcm::operator<< (std::ostream & *_os*, const GroupDict & *_val*) [inline]

References gdcm::GroupDict::GetAbbreviation(), gdcm::GroupDict::GetName(), and gdcm::GroupDict::Size().

24.1.4.19 std::ostream& gdcm::operator<< (std::ostream & *_os*, const IOD & *_val*) [inline]

24.1.4.20 std::ostream& gdcm::operator<< (std::ostream & *os*, const File & *val*) [inline]

References gdcm::File::GetHeader().

24.1.4.21 std::ostream& gdcm::operator<< (std::ostream & *_os*, const Usage & *val*) [inline]

References gdcm::Usage::GetUsageString().

24.1.4.22 std::ostream& gdcm::operator<< (std::ostream & *os*, const Sorter & *s*) [inline]

References gdcm::Sorter::Print().

24.1.4.23 std::ostream& gdcm::operator<< (std::ostream & *os*, const CSAHeaderDictEntry & *val*) [inline]

24.1.4.24 std::ostream& gdcm::operator<< (std::ostream & *os*, const Preamble & *val*) [inline]

24.1.4.25 std::ostream& gdcm::operator<< (std::ostream & *os*, const Dicts & *d*) [inline]

24.1.4.26 std::ostream& gdcm::operator<< (std::ostream & *_os*, const IODEntry & *_val*) [inline]

24.1.4.27 std::ostream& gdcm::operator<< (std::ostream & *_os*, const Macro & *_val*) [inline]

24.1.4.28 std::ostream& gdcm::operator<< (std::ostream & *os*, const CSAHeaderDict & *val*) [inline]

24.1.4.29 std::ostream& gdcm::operator<< (std::ostream & *os*, const PDBHeader & *d*) [inline]

References gdcm::PDBHeader::Print().

24.1.4.30 std::ostream& gdcm::operator<< (std::ostream & *os*, const CodeString & *str*) [inline]

24.1.4.31 std::ostream& gdcm::operator<< (std::ostream & *os*, const PrivateTag & *val*) [inline]

24.1.4.32 std::ostream& gdcm::operator<< (std::ostream & *_os*, const Module & *_val*) [inline]

24.1.4.33 std::ostream& gdcm::operator<< (std::ostream & *os*, const PhotometricInterpretation & *val*) [inline]

References gdcm::PhotometricInterpretation::GetPIString().

24.1.4.34 std::ostream& gdcm::operator<< (std::ostream & *os*, const Directory & *d*) [inline]

References gdcm::Directory::Print().

24.1.4.35 std::ostream& gdcm::operator<< (std::ostream & *os*, const Global & *g*) [inline]

24.1.4.36 std::ostream& gdcm::operator<< (std::ostream & *os*, const Object & *obj*) [inline]

References gdcm::Object::Print().

24.1.4.37 std::ostream& gdcm::operator<< (std::ostream & *os*, const BasicOffsetTable & *val*) [inline]

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLengthField.

24.1.4.38 std::ostream& gdcm::operator<< (std::ostream & *os*, const DictEntry & *val*) [inline]

24.1.4.39 std::ostream& gdcm::operator<< (std::ostream & *os*, const CSAElement & *val*) [inline]

References gdcm::CSAElement::DataField, gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), gdcm::CSAElement::KeyField, gdcm::CSAElement::NameField, gdcm::CSAElement::NoOfItemsField, gdcm::CSAElement::SyngoDTField, gdcm::CSAElement::ValueMultiplicityField, gdcm::VM::VM1, and gdcm::CSAElement::VRField.

24.1.4.40 std::ostream& gdcm::operator<< (std::ostream & *os*, const CSAHeader & *d*) [inline]

References gdcm::CSAHeader::Print().

24.1.4.41 std::ostream& gdcm::operator<< (std::ostream & *os*, const VL & *val*) [inline]

24.1.4.42 std::ostream& gdcm::operator<< (std::ostream & *_os*, const TransferSyntax & *ts*) [inline]

References gdcm::TransferSyntax::GetTSString().

24.1.4.43 std::ostream& gdcm::operator<< (std::ostream & *os*, const FileMetaInformation & *val*) [inline]

References gdcm::FileMetaInformation::GetPreamble(), and gdcm::DataSet::Print().

24.1.4.44 std::ostream& gdcm::operator<< (std::ostream & *_os*, const VM & *_val*) [inline]

References gdcm::VM::GetVMString().

24.1.4.45 std::ostream& gdcm::operator<< (std::ostream & *os*, const Fragment & *val*) [inline]

References gdcm::DataElement::TagField, gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLengthField.

24.1.4.46 std::ostream& gdcm::operator<< (std::ostream & *os*, const Scanner & *s*) [inline]

References gdcm::Scanner::Print().

24.1.4.47 std::ostream& gdcm::operator<< (std::ostream & *_os*, const MediaStorage & *ms*) [inline]

References gdcm::MediaStorage::GetMSString().

24.1.4.48 std::ostream& gdcm::operator<< (std::ostream & *os*, const Dict & *val*) [inline]

24.1.4.49 std::ostream& gdcm::operator<< (std::ostream & *os*, const PixelFormat & *pf*) [inline]

References gdcm::PixelFormat::Print().

24.1.4.50 std::ostream& gdcm::operator<< (std::ostream & *_os*, const VR & *val*) [inline]

References gdcm::VR::GetVRString().

24.1.4.51 std::ostream& gdcm::operator<< (std::ostream & _os, const UI & _val) [inline]

References gdcm::UI::Internal.

24.1.4.52 std::ostream& gdcm::operator<< (std::ostream & os, const DataElement & val) [inline]

References gdcm::Object::Print(), gdcm::DataElement::TagField, gdcm::DataElement::ValueField, gdcm::DataElement::ValueLengthField, and gdcm::DataElement::VRField.

24.1.4.53 std::ostream& gdcm::operator<< (std::ostream & _os, const Tag & _val) [inline]

24.1.4.54 std::ostream& gdcm::operator<< (std::ostream & os, const DataSet & val) [inline]

References gdcm::DataSet::Print().

24.1.4.55 std::ostream& gdcm::operator<< (std::ostream & os, const Item & val) [inline]

References gdcm::DataSet::Print(), gdcm::DataElement::TagField, and gdcm::DataElement::ValueLengthField.

24.1.4.56 std::ostream& gdcm::operator<< (std::ostream & os, const PrivateDict & val) [inline]

24.1.4.57 std::ostream& gdcm::operator<< (std::ostream & _os, const UIDs & uid) [inline]

References gdcm::UIDs::GetName(), and gdcm::UIDs::GetString().

24.1.4.58 bool gdcm::operator== (const CodeString & ref, const CodeString & cs) [inline]

24.1.4.59 template<char TDelimiter, unsigned int TMaxLength, char TPadChar> std::istream& gdcm::operator>> (std::istream & is, String< TDelimiter, TMaxLength, TPadChar > & ms) [inline]

24.1.4.60 std::istream& gdcm::operator>> (std::istream & in, ignore_char const & ic) [inline]

References gdcm::ignore_char::m_char.

24.1.4.61 std::istream& gdcm::operator>> (std::istream & _is, Tag & _val) [inline]

References gdcm::Tag::SetElement(), and gdcm::Tag::SetGroup().

24.1.4.62 template<typename Float > std::string gdcm::to_string (Float data)

Referenced by gdcm::EncodingImplementation< VR::VRASCII >::Write().

24.1.4.63 gdcm::TYPETOENCODING (SQ , VRBINARY , unsigned char)

24.1.5 Variable Documentation

24.1.5.1 Global gdcm::GlobalInstance [static]

24.1.5.2 gdcm::VRBINARY

24.2 gdcm::network Namespace Reference

Classes

- class AAbortPDU

  AAbortPDU Table 9-26 A-ABORT PDU FIELDS.
- class AAssociateACPDU

  AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.
- class AAssociateRJPDU

  AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.
- class AAssociateRQPDU

  AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.
- class AbstractSyntax

  AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.
- class ApplicationContext

  ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)
- class AReleaseRPPDU

  AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.
- class AReleaseRQPDU

  AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.
- class ARTIMTimer

  ARTIMTimer This file contains the code for the ARTIM timer.
- class AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class BaseCompositeMessage

  BaseCompositeMessage The Composite events described in section 3.7-2009 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2009 of the standard, and then fill in appropriate values in their datasets.
- class BasePDU

  BasePDU base class for PDUs.
- class CEchoRQ

  CEchoRQ this file defines the messages for the cecho action.
- class CEchoRSP
- class CFind
- class CFindCancelRQ
- class CFindRQ
- class CFindRSP
- class CMoveCancelRq
- class CMoveRQ

  CMoveRQ this file defines the messages for the cmove action.
- class CMoveRSP

  CMoveRSP this file defines the messages for the cmove action.
- class CompositeMessageFactory

CompositeMessageFactory This class constructs PDataPDUs, but that have been specifically constructed for the composite DICOM services (C-Echo, C-Find, C-Get, C-Move, and C-Store). It will also handle parsing the incoming data to determine which of the CompositePDUs the incoming data is, and so therefore allowing the scu to determine what to do with incoming data (if acting as a storescp server, for instance).

- class CStoreRQ

  CStoreRQ this file defines the messages for the cecho action.
- class CStoreRSP
- class DIMSE

  DIMSE PS 3.7 - 2009 Annex E Command Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS Table E.1-1 COMMAND FIELDS (PART 1)
- class ImplementationClassUIDSub

  ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE--RQ)
- class ImplementationUIDSub

  ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)
- class ImplementationVersionNameSub

  ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class MaximumLengthSub

  MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class PDataTFPDU

  PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.
- class PDUFactory

  PDUFactory basically, given an initial byte, construct the appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.
- class PresentationContextAC

  PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.
- class PresentationContextRQ

  PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.
- class PresentationDataValue

  PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.
- class TableRow
- class TransferSyntaxSub

  TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.
- struct Transition
- class ULAction

  ULAction A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.
- class ULActionAA1
- class ULActionAA2
- class ULActionAA3
- class ULActionAA4
- class ULActionAA5
- class ULActionAA6
- class ULActionAA7
- class ULActionAA8
- class ULActionAE1
- class ULActionAE2
- class ULActionAE3
- class ULActionAE4

- class ULActionAE5
- class ULActionAE6
- class ULActionAE7
- class ULActionAE8
- class ULActionAR1
- class ULActionAR10
- class ULActionAR2
- class ULActionAR3
- class ULActionAR4
- class ULActionAR5
- class ULActionAR6
- class ULActionAR7
- class ULActionAR8
- class ULActionAR9
- class ULActionDT1
- class ULActionDT2
- class ULBasicCallback
- class ULConnection

  ULConnection This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

- class ULConnectionCallback
- class ULConnectionInfo

  ULConnectionInfo this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

- class ULConnectionManager

  ULConnectionManager The ULConnectionManager performs actions on the ULConnection given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

- class ULEvent

  ULEvent base class for network events.

- class ULTransitionTable

  ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

- class ULWritingCallback
- class UserInformation

  UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

Enumerations

- enum EEventID {
  eAASSOCIATERequestLocalUser = 0,
  eTransportConnConfirmLocal,
  eASSOCIATE_ACPDUreceived,
  eASSOCIATE_RJPDUreceived,
  eTransportConnIndicLocal,
  eAASSOCIATE_RQPDUreceived,
  eAASSOCIATEresponseAccept,
  eAASSOCIATEresponseReject,
  ePDATArequest,
  ePDATATFPDU,
  eARELEASERequest,
  eARELEASE_RQPDUReceivedOpen,
  eARELEASE_RPPDUReceived,
  eARELEASEResponse,
  eAABORTRequest,
  eAABORTPDUReceivedOpen,
  eTransportConnectionClosed,
  eARTIMTimerExpired,
  eUnrecognizedPDUReceived,
  eEventDoesNotExist }
- enum EStateID {
  eStaDoesNotExist = 0,
  eSta1Idle = 1,
  eSta2Open = 2,
  eSta3WaitLocalAssoc = 4,
  eSta4LocalAssocDone = 8,
  eSta5WaitRemoteAssoc = 16,
  eSta6TransferReady = 32,
  eSta7WaitRelease = 64,
  eSta8WaitLocalRelease = 128,
  eSta9ReleaseCollisionRqLocal = 256,
  eSta10ReleaseCollisionAc = 512,
  eSta11ReleaseCollisionRq = 1024,
  eSta12ReleaseCollisionAcLocal = 2048,
  eSta13AwaitingClose = 4096 }

Functions

- int GetStateIndex (EStateID inState)

Variables

- const int cMaxEventID = eEventDoesNotExist
- const int cMaxStateID = 13

24.2.1 Enumeration Type Documentation

24.2.1.1 enum gdcm::network::EEventID

Enumerator:

eAASSOCIATERequestLocalUser
eTransportConnConfirmLocal
eASSOCIATE_ACPDUreceived
eASSOCIATE_RJPDUreceived
eTransportConnIndicLocal
eAASSOCIATE_RQPDUreceived
eAASSOCIATEresponseAccept
eAASSOCIATEresponseReject
ePDATArequest
ePDATATFPDU
eARELEASERequest
eARELEASE_RQPDUReceivedOpen
eARELEASE_RPPDUReceived
eARELEASEResponse
eAABORTRequest
eAABORTPDUReceivedOpen
eTransportConnectionClosed
eARTIMTimerExpired
eUnrecognizedPDUReceived
eEventDoesNotExist

24.2.1.2 enum gdcm::network::EStateID

Each network connection will be in a particular state at any given time. Those states have IDs as described in the standard ps3.8-2009, roughly 1-13. This enumeration lists those states. The actual ULState class will contain more information about transitions to other states.

name and date: 16 sept 2010 mmr

Enumerator:

eStaDoesNotExist
eSta1Idle
eSta2Open
eSta3WaitLocalAssoc
eSta4LocalAssocDone
eSta5WaitRemoteAssoc
eSta6TransferReady
eSta7WaitRelease
eSta8WaitLocalRelease
eSta9ReleaseCollisionRqLocal
eSta10ReleaseCollisionAc
eSta11ReleaseCollisionRq
eSta12ReleaseCollisionAcLocal
eSta13AwaitingClose

24.2.2 Function Documentation

24.2.2.1 int gdcm::network::GetStateIndex (EStateID *inState*) [inline]

References eSta10ReleaseCollisionAc, eSta11ReleaseCollisionRq, eSta12ReleaseCollisionAcLocal, eSta13AwaitingClose, eSta1Idle, eSta2Open, eSta3WaitLocalAssoc, eSta4LocalAssocDone, eSta5WaitRemoteAssoc, eSta6TransferReady, eSta7WaitRelease, eSta8WaitLocalRelease, eSta9ReleaseCollisionRqLocal, and eStaDoesNotExist.

24.2.3 Variable Documentation

24.2.3.1 const int gdcm::network::cMaxEventID = eEventDoesNotExist

24.2.3.2 const int gdcm::network::cMaxStateID = 13

24.3 gdcm::SegmentHelper Namespace Reference

Classes

- struct BasicCodedEntry

  This structure defines a basic coded entry with all of its attributes.

24.4 gdcm::terminal Namespace Reference

Class for Terminal Allow one to print in color in a shell.

Enumerations

- enum Attribute {
  reset = 0,
  bright = 1,
  dim = 2,
  underline = 3,
  blink = 5,
  reverse = 7,
  hidden = 8 }
- enum Color {
  black = 0,
  red,
  green,
  yellow,
  blue,
  magenta,
  cyan,
  white }
- enum Mode {
  CONSOLE = 0,
  VT100 }

Functions

- GDCM_EXPORT std::string setattribute (Attribute att)
- GDCM_EXPORT std::string setbgcolor (Color c)
- GDCM_EXPORT std::string setfgcolor (Color c)
- GDCM_EXPORT void setmode (Mode m)

24.4.1 Detailed Description

Class for Terminal Allow one to print in color in a shell.

- support VT100 compatible shell
- win32 console

24.4.2 Enumeration Type Documentation

24.4.2.1 enum gdcm::terminal::Attribute

Enumerator:

reset
bright
dim
underline
blink
reverse
hidden

24.4.2.2 enum gdcm::terminal::Color

Enumerator:

black
red
green
yellow
blue
magenta
cyan
white

24.4.2.3 enum gdcm::terminal::Mode

Enumerator:

CONSOLE
VT100

24.4.3 Function Documentation

24.4.3.1 GDCM_EXPORT std::string gdcm::terminal::setattribute (Attribute *att*)

24.4.3.2 GDCM_EXPORT std::string gdcm::terminal::setbgcolor (Color *c*)

24.4.3.3 GDCM_EXPORT std::string gdcm::terminal::setfgcolor (Color *c*)

24.4.3.4 GDCM_EXPORT void gdcm::terminal::setmode (Mode *m*)

24.5 itk Namespace Reference

Classes

- class GDCMImageIO2

  ImageIO class for reading and writing DICOM V3.0 and ACR/NEMA (V1.0 & V2.0) images This class is only an adaptor to the gdcm library (currently gdcm 2.0 is used):

Chapter 25

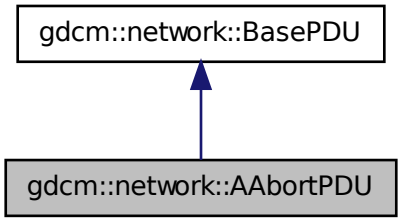
Class Documentation

25.1 gdcm::network::AAbortPDU Class Reference

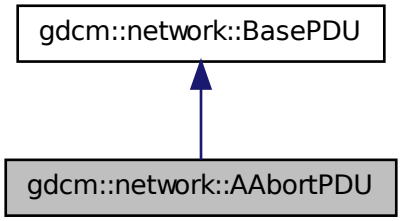
AAbortPDU Table 9-26 A-ABORT PDU FIELDS.

```
#include <gdcmAAbortPDU.h>
```

Inheritance diagram for gdcm::network::AAbortPDU:



Collaboration diagram for gdcm::network::AAbortPDU:



Public Member Functions

- AAbortPDU ()
- bool IsLastFragment () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.1.1 Detailed Description

AAbortPDU Table 9-26 A-ABORT PDU FIELDS.

25.1.2 Constructor & Destructor Documentation

25.1.2.1 gdcm::network::AAbortPDU::AAbortPDU ( )

25.1.3 Member Function Documentation

25.1.3.1 bool gdcm::network::AAbortPDU::IsLastFragment ( ) const [inline], [virtual]

Implements gdcm::network::BasePDU.

25.1.3.2 void gdcm::network::AAbortPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

25.1.3.3 std::istream& gdcm::network::AAbortPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

25.1.3.4 size_t gdcm::network::AAbortPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

25.1.3.5 const std::ostream& gdcm::network::AAbortPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

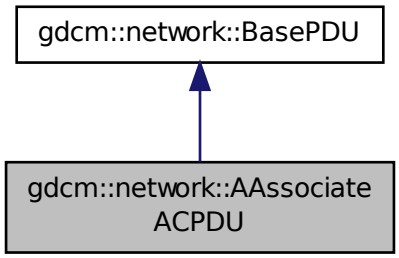
- gdcmAAbortPDU.h

25.2 gdcm::network::AAssociateACPDU Class Reference

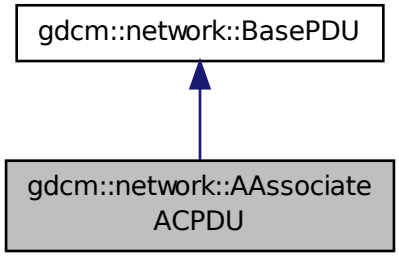
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.

`#include <gdcmAAssociateACPDU.h>`

Inheritance diagram for gdcm::network::AAssociateACPDU:



Collaboration diagram for gdcm::network::AAssociateACPDU:



Public Types

- typedef std::vector < PresentationContextAC > ::size_type SizeType

Public Member Functions

- AAssociateACPDU ()
- void AddPresentationContextAC (PresentationContextAC const &pcac)
- SizeType GetNumberOfPresentationContextAC () const
- const PresentationContextAC & GetPresentationContextAC (SizeType i)
- const UserInformation & GetUserInformation () const
- void InitFromRQ (AAssociateRQPDU const &rqpdu)
- bool IsLastFragment () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- SizeType Size () const
- const std::ostream & Write (std::ostream &os) const

Protected Member Functions

- void SetCalledAETitle (const char calledaetitle[16])
- void SetCallingAETitle (const char callingaetitle[16])

Friends

- class AAssociateRQPDU

25.2.1 Detailed Description

AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.

25.2.2 Member Typedef Documentation

25.2.2.1 typedef std::vector<PresentationContextAC>::size_type gdcm::network::AAssociateACPDU::SizeType

25.2.3 Constructor & Destructor Documentation

25.2.3.1 gdcm::network::AAssociateACPDU::AAssociateACPDU ( )

25.2.4 Member Function Documentation

25.2.4.1 void gdcm::network::AAssociateACPDU::AddPresentationContextAC (PresentationContextAC const & *pcac*)

25.2.4.2 SizeType gdcm::network::AAssociateACPDU::GetNumberOfPresentationContextAC ( ) const [inline]

25.2.4.3 const PresentationContextAC& gdcm::network::AAssociateACPDU::GetPresentationContextAC (SizeType *i*) [inline]

25.2.4.4 const UserInformation& gdcm::network::AAssociateACPDU::GetUserInformation ( ) const [inline]

25.2.4.5 void gdcm::network::AAssociateACPDU::InitFromRQ (AAssociateRQPDU const & *rqpdu*)

25.2.4.6 bool gdcm::network::AAssociateACPDU::IsLastFragment ( ) const [inline], [virtual]

Implements gdcm::network::BasePDU.

25.2.4.7 void gdcm::network::AAssociateACPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

25.2.4.8 std::istream& gdcm::network::AAssociateACPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

25.2.4.9 void gdcm::network::AAssociateACPDU::SetCalledAETitle (const char *calledaetitle[16]*) [protected]

25.2.4.10 void gdcm::network::AAssociateACPDU::SetCallingAETitle (const char *callingaetitle[16]*) [protected]

25.2.4.11 SizeType gdcm::network::AAssociateACPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

25.2.4.12 const std::ostream& gdcm::network::AAssociateACPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

25.2.5 Friends And Related Function Documentation

25.2.5.1 friend class AAssociateRQPDU [friend]

The documentation for this class was generated from the following file:

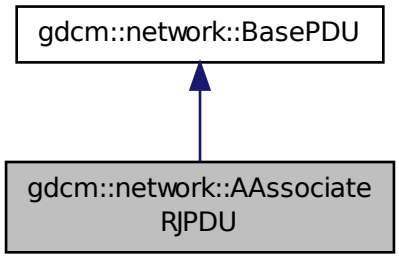
- gdcmAAssociateACPDU.h

25.3 gdcm::network::AAssociateRJPDU Class Reference

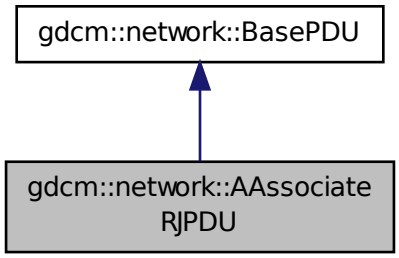
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

`#include <gdcmAAssociateRJPDU.h>`

Inheritance diagram for gdcm::network::AAssociateRJPDU:



Collaboration diagram for gdcm::network::AAssociateRJPDU:



Public Member Functions

- AAssociateRJPDU ()

- bool IsLastFragment () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.3.1 Detailed Description

AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

25.3.2 Constructor & Destructor Documentation

25.3.2.1 gdcm::network::AAssociateRJPDU::AAssociateRJPDU ( )

25.3.3 Member Function Documentation

25.3.3.1 bool gdcm::network::AAssociateRJPDU::IsLastFragment ( ) const [inline], [virtual]

Implements gdcm::network::BasePDU.

25.3.3.2 void gdcm::network::AAssociateRJPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

25.3.3.3 std::istream& gdcm::network::AAssociateRJPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

25.3.3.4 size_t gdcm::network::AAssociateRJPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

25.3.3.5 const std::ostream& gdcm::network::AAssociateRJPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

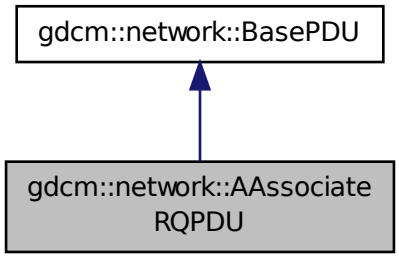
- gdcmAAssociateRJPDU.h

25.4 gdcm::network::AAssociateRQPDU Class Reference

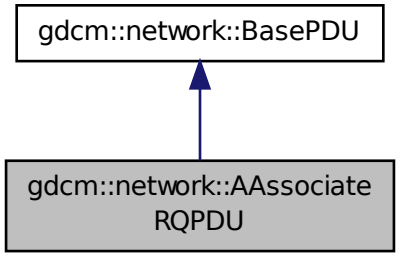
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

```
#include <gdcmAAssociateRQPDU.h>
```

Inheritance diagram for gdcm::network::AAssociateRQPDU:



Collaboration diagram for gdcm::network::AAssociateRQPDU:



Public Types

- typedef std::vector
  < PresentationContextRQ > PresentationContextArrayType
- typedef std::vector
  < PresentationContextRQ >
  ::size_type SizeType

Public Member Functions

- AAssociateRQPDU ()
- AAssociateRQPDU (const AAssociateRQPDU &)
- void AddPresentationContext (PresentationContextRQ const &pc)
- std::string GetCalledAETitle () const
- std::string GetCallingAETitle () const

- SizeType GetNumberOfPresentationContext () const
- PresentationContextRQ const & GetPresentationContext (SizeType i) const
- const PresentationContextRQ * GetPresentationContextByAbstractSyntax (AbstractSyntax const &as) const
- const PresentationContextRQ * GetPresentationContextByID (uint8_t i) const
- PresentationContextArrayType const & GetPresentationContexts ()
- bool IsLastFragment () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetCalledAETitle (const char calledaetitle[16])

  Set the Called AE Title.
- void SetCallingAETitle (const char callingaetitle[16])

  Set the Calling AE Title.
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

Static Public Member Functions

- static bool IsAETitleValid (const char title[16])

  Check whether or not the title is a valid AE title.

25.4.1 Detailed Description

AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

25.4.2 Member Typedef Documentation

25.4.2.1 typedef std::vector<PresentationContextRQ> gdcm::network::AAssociateRQPDU::PresentationContextArrayType

25.4.2.2 typedef std::vector<PresentationContextRQ>::size_type gdcm::network::AAssociateRQPDU::SizeType

25.4.3 Constructor & Destructor Documentation

25.4.3.1 gdcm::network::AAssociateRQPDU::AAssociateRQPDU ( )

25.4.3.2 gdcm::network::AAssociateRQPDU::AAssociateRQPDU (const AAssociateRQPDU &) [inline]

25.4.4 Member Function Documentation

25.4.4.1 void gdcm::network::AAssociateRQPDU::AddPresentationContext (PresentationContextRQ const & *pc*)

25.4.4.2 std::string gdcm::network::AAssociateRQPDU::GetCalledAETitle ( ) const [inline]

25.4.4.3 std::string gdcm::network::AAssociateRQPDU::GetCallingAETitle ( ) const [inline]

25.4.4.4 SizeType gdcm::network::AAssociateRQPDU::GetNumberOfPresentationContext ( ) const [inline]

25.4.4.5 PresentationContextRQ const& gdcm::network::AAssociateRQPDU::GetPresentationContext (SizeType *i*) const [inline]

25.4.4.6 const PresentationContextRQ* gdcm::network::AAssociateRQPDU::GetPresentationContextByAbstractSyntax (AbstractSyntax const & *as*) const

25.4.4.7 const PresentationContextRQ* gdcm::network::AAssociateRQPDU::GetPresentationContextByID (uint8_t *i*) const

25.4.4.8 PresentationContextArrayType const& gdcm::network::AAssociateRQPDU::GetPresentationContexts ( ) [inline]

25.4.4.9 static bool gdcm::network::AAssociateRQPDU::IsAETitleValid (const char *title[16]*) [static]

Check whether or not the title is a valid AE title.

25.4.4.10 bool gdcm::network::AAssociateRQPDU::IsLastFragment ( ) const [inline], [virtual]

Implements gdcm::network::BasePDU.

25.4.4.11 void gdcm::network::AAssociateRQPDU::Print (std::ostream & *os*) const [virtual]

This function will initialize an AAssociateACPDU from the fields in the AAssociateRQPDU structure

Implements gdcm::network::BasePDU.

25.4.4.12 std::istream& gdcm::network::AAssociateRQPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

25.4.4.13 void gdcm::network::AAssociateRQPDU::SetCalledAETitle (const char *calledaetitle[16]*)

Set the Called AE Title.

25.4.4.14 void gdcm::network::AAssociateRQPDU::SetCallingAETitle (const char *callingaetitle[16]*)

Set the Calling AE Title.

25.4.4.15 size_t gdcm::network::AAssociateRQPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

25.4.4.16 const std::ostream& gdcm::network::AAssociateRQPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

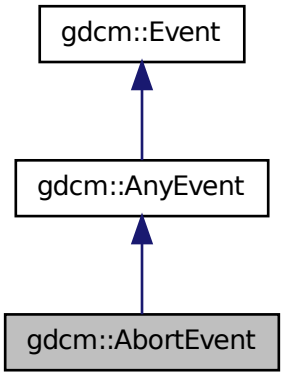
The documentation for this class was generated from the following file:

- gdcmAAssociateRQPDU.h

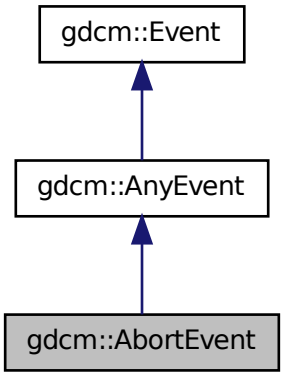
25.5 gdcm::AbortEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::AbortEvent:



Collaboration diagram for gdcm::AbortEvent:



The documentation for this class was generated from the following file:

- gdcmEvent.h

25.6 gdcm::network::AbstractSyntax Class Reference

AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

```
#include <gdcmAbstractSyntax.h>
```

Public Member Functions

- AbstractSyntax ()
- DataElement GetAsDataElement () const
- const char ∗ GetName () const
- bool operator== (const AbstractSyntax &as) const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetName (const char ∗name)
- void SetNameFromUID (UIDs::TSName tsname)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.6.1 Detailed Description

AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

25.6.2 Constructor & Destructor Documentation

25.6.2.1 gdcm::network::AbstractSyntax::AbstractSyntax ( )

25.6.3 Member Function Documentation

25.6.3.1 DataElement gdcm::network::AbstractSyntax::GetAsDataElement ( ) const

25.6.3.2 const char∗ gdcm::network::AbstractSyntax::GetName ( ) const [inline]

25.6.3.3 bool gdcm::network::AbstractSyntax::operator== (const AbstractSyntax & *as*) const [inline]

25.6.3.4 void gdcm::network::AbstractSyntax::Print (std::ostream & *os*) const

25.6.3.5 std::istream& gdcm::network::AbstractSyntax::Read (std::istream & *is*)

25.6.3.6 void gdcm::network::AbstractSyntax::SetName (const char ∗ *name*) [inline]

25.6.3.7 void gdcm::network::AbstractSyntax::SetNameFromUID (UIDs::TSName *tsname*)

25.6.3.8 size_t gdcm::network::AbstractSyntax::Size ( ) const

25.6.3.9 const std::ostream& gdcm::network::AbstractSyntax::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

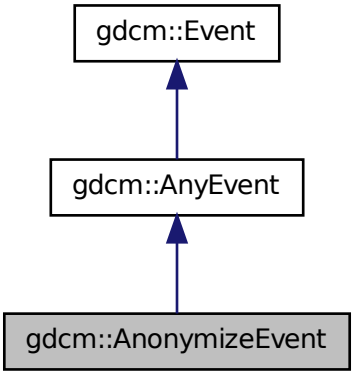
- gdcmAbstractSyntax.h

25.7 gdcm::AnonymizeEvent Class Reference

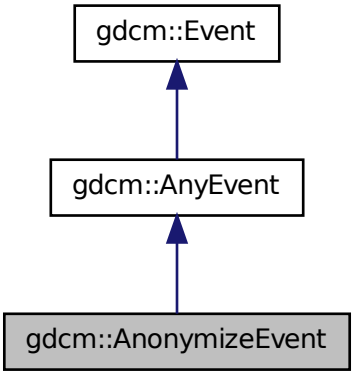
AnonymizeEvent Special type of event triggered during the Anonymization process.

```
#include <gdcmAnonymizeEvent.h>
```

Inheritance diagram for gdcm::AnonymizeEvent:



Collaboration diagram for gdcm::AnonymizeEvent:



Public Types

- typedef AnonymizeEvent Self

- typedef AnyEvent Superclass

Public Member Functions

- AnonymizeEvent (Tag const &tag=0)
- AnonymizeEvent (const Self &s)
- virtual ∼AnonymizeEvent ()
- virtual bool CheckEvent (const ::gdcm::Event ∗e) const
- virtual const char ∗ GetEventName () const
- Tag const & GetTag () const
- virtual ::gdcm::Event ∗ MakeObject () const
- void SetTag (const Tag &t)

25.7.1 Detailed Description

AnonymizeEvent Special type of event triggered during the Anonymization process.

See also

Anonymizer

25.7.2 Member Typedef Documentation

25.7.2.1 typedef AnonymizeEvent gdcm::AnonymizeEvent::Self

25.7.2.2 typedef AnyEvent gdcm::AnonymizeEvent::Superclass

25.7.3 Constructor & Destructor Documentation

25.7.3.1 gdcm::AnonymizeEvent::AnonymizeEvent (Tag const & *tag* = 0) [inline]

25.7.3.2 virtual gdcm::AnonymizeEvent::∼AnonymizeEvent ( ) [inline], [virtual]

25.7.3.3 gdcm::AnonymizeEvent::AnonymizeEvent (const Self & *s*) [inline]

25.7.4 Member Function Documentation

25.7.4.1 virtual bool gdcm::AnonymizeEvent::CheckEvent (const ::gdcm::Event ∗ *e*) const [inline], [virtual]

25.7.4.2 virtual const char∗ gdcm::AnonymizeEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

25.7.4.3 Tag const& gdcm::AnonymizeEvent::GetTag ( ) const [inline]

25.7.4.4 virtual ::gdcm::Event∗ gdcm::AnonymizeEvent::MakeObject ( ) const [inline], [virtual]

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

25.7.4.5 void gdcm::AnonymizeEvent::SetTag (const Tag & *t*) [inline]

The documentation for this class was generated from the following file:

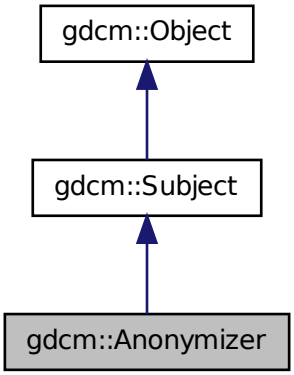
- gdcmAnonymizeEvent.h

25.8 gdcm::Anonymizer Class Reference

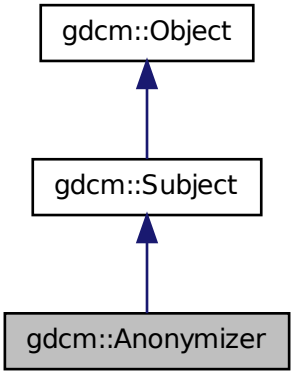
Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:

`#include <gdcmAnonymizer.h>`

Inheritance diagram for gdcm::Anonymizer:



Collaboration diagram for gdcm::Anonymizer:



Public Member Functions

- Anonymizer ()
- ∼Anonymizer ()
- bool BasicApplicationLevelConfidentialityProfile (bool deidentify=true)
- bool Empty (Tag const &t)
- const CryptographicMessageSyntax ∗ GetCryptographicMessageSyntax () const
- File & GetFile ()
- bool Remove (Tag const &t)

  remove a tag (even a SQ can be removed)
- bool RemoveGroupLength ()

  Main function that loop over all elements and remove group length.
- bool RemovePrivateTags ()

  Main function that loop over all elements and remove private tags.
- bool RemoveRetired ()

  Main function that loop over all elements and remove retired element.
- bool Replace (Tag const &t, const char ∗value)
- bool Replace (Tag const &t, const char ∗value, VL const &vl)
- void SetCryptographicMessageSyntax (CryptographicMessageSyntax ∗cms)

  Set/Get CMS key that will be used to encrypt the dataset within BasicApplicationLevelConfidentialityProfile.
- void SetFile (const File &f)

  Set/Get File.

Static Public Member Functions

- static std::vector< Tag > GetBasicApplicationLevelConfidentialityProfileAttributes ()

  Return the list of Tag that will be considered when anonymizing a DICOM file.
- static SmartPointer< Anonymizer > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- bool BALCPProtect (DataSet &ds, Tag const &tag, const IOD &iod)
- bool CanEmptyTag (Tag const &tag, const IOD &iod) const
- void RecurseDataSet (DataSet &ds)

25.8.1 Detailed Description

Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:

- Full (irreversible) anonymizer (aka dumb mode)
- reversible de-identifier/re-identifier (aka smart mode). This implements the Basic Application Level Confidentiality Profile, DICOM PS 3.15-2009

1. dumb mode This is a dumb anonymizer implementation. All it allows user is simple operation such as:

Tag based functions:

- complete removal of DICOM attribute (Remove)
- make a tag empty, ie make it's length 0 (Empty)
- replace with another string-based value (Replace)

DataSet based functions:

- Remove all group length attribute from a DICOM dataset (Group Length element are deprecated, DICOM 2008)
- Remove all private attributes
- Remove all retired attributes

All function calls actually execute the user specified request. Previous implementation were calling a general Anonymize function but traversing a std::set is O(n) operation, while a simple user specified request is O(log(n)) operation. So 'm' user interaction is O(m$*$log(n)) which is $<$ O(n) complexity.

1. smart mode this mode implements the Basic Application Level Confidentiality Profile (DICOM PS 3.15-2008) In this case, it is extremely important to use the same gdcm::Anonymizer class when anonymizing a FileSet. Once the gdcm::Anonymizer is destroyed its memory of known (already processed) UIDs will be lost. which will make the anonymizer behaves incorrectly for attributes such as Series UID Study UID where user want some consistency. When attribute is Type 1 / Type 1C, a dummy generator will take in the existing value and produce a dummy value (a sha1 representation). sha1 algorithm is considered to be cryptographically strong (compared to md5sum) so that we meet the following two conditions:
   - Produce the same dummy value for the same input value
   - do not provide an easy way to retrieve the original value from the sha1 generated value

This class implement the Subject/Observer pattern trigger the following event:

- AnonymizeEvent
- IterationEvent
- StartEvent
- EndEvent

See also

CryptographicMessageSyntax

Examples:

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, and EncapsulateFileInRawData.cxx.

25.8.2 Constructor & Destructor Documentation

25.8.2.1 gdcm::Anonymizer::Anonymizer ( ) [inline]

25.8.2.2 gdcm::Anonymizer::~Anonymizer ( )

25.8.3 Member Function Documentation

25.8.3.1 bool gdcm::Anonymizer::BALCPProtect (DataSet & *ds,* Tag const & *tag,* const IOD & *iod*) [protected]

25.8.3.2 bool gdcm::Anonymizer::BasicApplicationLevelConfidentialityProfile (bool *deidentify* = `true`)

PS 3.15 / E.1.1 De-Identifier An Application may claim conformance to the Basic Application Level Confidentiality Profile as a deidentifier if it protects all Attributes that might be used by unauthorized entities to identify the patient. NOT THREAD SAFE

25.8.3.3 bool gdcm::Anonymizer::CanEmptyTag (Tag const & *tag,* const IOD & *iod*) const [protected]

25.8.3.4 bool gdcm::Anonymizer::Empty (Tag const & *t*)

Make Tag t empty (if not found tag will be created) Warning: does not handle SQ element

Examples:

CreateJPIPDataSet.cxx.

25.8.3.5 static std::vector<Tag> gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes ( ) [static]

Return the list of Tag that will be considered when anonymizing a DICOM file.

Examples:

GenFakeIdentifyFile.cxx, and TraverseModules.cxx.

25.8.3.6 const CryptographicMessageSyntax* gdcm::Anonymizer::GetCryptographicMessageSyntax ( ) const

25.8.3.7 File& gdcm::Anonymizer::GetFile ( ) [inline]

25.8.3.8 static SmartPointer<Anonymizer> gdcm::Anonymizer::New ( ) [inline],[static]

for wrapped language: instantiate a reference counted object

25.8.3.9 void gdcm::Anonymizer::RecurseDataSet (DataSet & *ds*) [protected]

25.8.3.10 bool gdcm::Anonymizer::Remove (Tag const & *t*)

remove a tag (even a SQ can be removed)

25.8.3.11 bool gdcm::Anonymizer::RemoveGroupLength ( )

Main function that loop over all elements and remove group length.

Examples:

ClinicalTrialAnnotate.cxx.

25.8.3.12 bool gdcm::Anonymizer::RemovePrivateTags ( )

Main function that loop over all elements and remove private tags.

Examples:

ClinicalTrialAnnotate.cxx.

25.8.3.13 bool gdcm::Anonymizer::RemoveRetired ( )

Main function that loop over all elements and remove retired element.

25.8.3.14 bool gdcm::Anonymizer::Replace (Tag const & *t,* const char * *value*)

Replace tag with another value, if tag is not found it will be created: WARNING: this function can only execute if tag is a VRASCII

Examples:

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, and EncapsulateFileInRawData.cxx.

25.8.3.15 bool gdcm::Anonymizer::Replace (Tag const & *t,* const char * *value,* VL const & *vl*)

when the value contains \0, it is a good idea to specify the length. This function is required when dealing with VRBINARY tag

25.8.3.16 void gdcm::Anonymizer::SetCryptographicMessageSyntax (CryptographicMessageSyntax * *cms*)

Set/Get CMS key that will be used to encrypt the dataset within BasicApplicationLevelConfidentialityProfile.

25.8.3.17 void gdcm::Anonymizer::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, and EncapsulateFileInRawData.cxx.

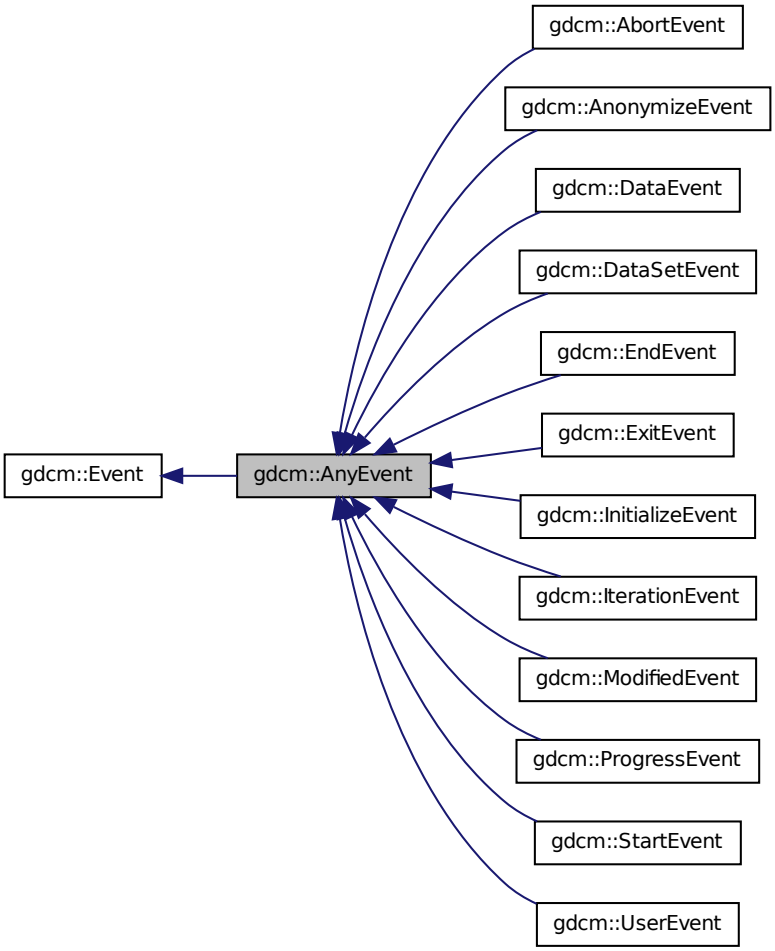
The documentation for this class was generated from the following file:

- gdcmAnonymizer.h

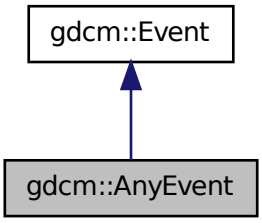
25.9 gdcm::AnyEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::AnyEvent:



Collaboration diagram for gdcm::AnyEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmEvent.h

25.10 gdcm::network::ApplicationContext Class Reference

ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

```
#include <gdcmApplicationContext.h>
```

Public Member Functions

- ApplicationContext ()
- const char ∗ GetName () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetName (const char ∗name)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.10.1 Detailed Description

ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

25.10.2 Constructor & Destructor Documentation

25.10.2.1 gdcm::network::ApplicationContext::ApplicationContext ( )

25.10.3 Member Function Documentation

25.10.3.1 const char* gdcm::network::ApplicationContext::GetName ( ) const [inline]

25.10.3.2 void gdcm::network::ApplicationContext::Print (std::ostream & *os*) const

25.10.3.3 std::istream& gdcm::network::ApplicationContext::Read (std::istream & *is*)

25.10.3.4 void gdcm::network::ApplicationContext::SetName (const char * *name*) [inline]

25.10.3.5 size_t gdcm::network::ApplicationContext::Size ( ) const

25.10.3.6 const std::ostream& gdcm::network::ApplicationContext::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

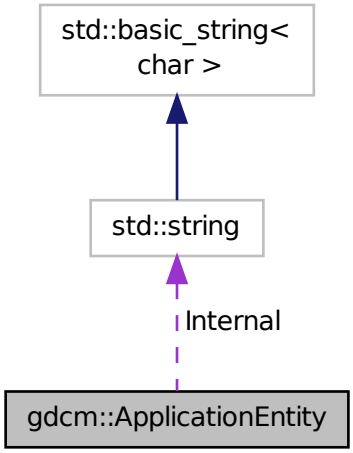
- gdcmApplicationContext.h

25.11 gdcm::ApplicationEntity Class Reference

ApplicationEntity.

`#include <gdcmApplicationEntity.h>`

Collaboration diagram for gdcm::ApplicationEntity:



Public Member Functions

- bool IsValid () const

- void Print (std::ostream &os) const
- void SetBlob (const std::vector< char > &v)
- void Squeeze ()

Public Attributes

- std::string Internal

Static Public Attributes

- static const unsigned int MaxLength = 16
- static const unsigned int MaxNumberOfComponents = 1
- static const char Padding = ' '
- static const char Separator = ' '

25.11.1 Detailed Description

ApplicationEntity.

- AE Application Entity
- A string of characters that identifies an Application Entity with leading and trailing spaces (20H) being non-significant. A value consisting solely of spaces shall not be used.
- Default Character Repertoire excluding character code 5CH (the BACKSLASH \ in ISO-IR 6), and control characters LF, FF, CR and ESC.
- 16 bytes maximum

25.11.2 Member Function Documentation

25.11.2.1 bool gdcm::ApplicationEntity::IsValid ( ) const [inline]

25.11.2.2 void gdcm::ApplicationEntity::Print (std::ostream & *os*) const [inline]

25.11.2.3 void gdcm::ApplicationEntity::SetBlob (const std::vector< char > & *v*) [inline]

25.11.2.4 void gdcm::ApplicationEntity::Squeeze ( ) [inline]

25.11.3 Member Data Documentation

25.11.3.1 std::string gdcm::ApplicationEntity::Internal

25.11.3.2 const unsigned int gdcm::ApplicationEntity::MaxLength = 16 [static]

25.11.3.3 const unsigned int gdcm::ApplicationEntity::MaxNumberOfComponents = 1 [static]

25.11.3.4 const char gdcm::ApplicationEntity::Padding = ' ' [static]

25.11.3.5 const char gdcm::ApplicationEntity::Separator = ' ' [static]

The documentation for this class was generated from the following file:

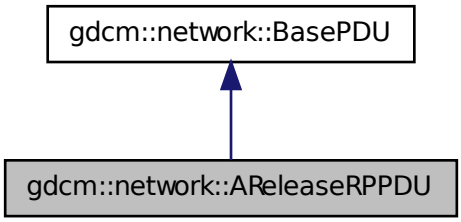
- gdcmApplicationEntity.h

25.12 gdcm::network::AReleaseRPPDU Class Reference

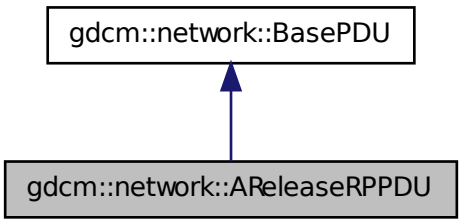
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

`#include <gdcmAReleaseRPPDU.h>`

Inheritance diagram for gdcm::network::AReleaseRPPDU:



Collaboration diagram for gdcm::network::AReleaseRPPDU:



Public Member Functions

- AReleaseRPPDU ()
- bool IsLastFragment () const
- void Print (std::ostream &os) const

- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.12.1 Detailed Description

AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

25.12.2 Constructor & Destructor Documentation

25.12.2.1 gdcm::network::AReleaseRPPDU::AReleaseRPPDU ( )

25.12.3 Member Function Documentation

25.12.3.1 bool gdcm::network::AReleaseRPPDU::IsLastFragment ( ) const [inline],[virtual]

Implements gdcm::network::BasePDU.

25.12.3.2 void gdcm::network::AReleaseRPPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

25.12.3.3 std::istream& gdcm::network::AReleaseRPPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

25.12.3.4 size_t gdcm::network::AReleaseRPPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

25.12.3.5 const std::ostream& gdcm::network::AReleaseRPPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

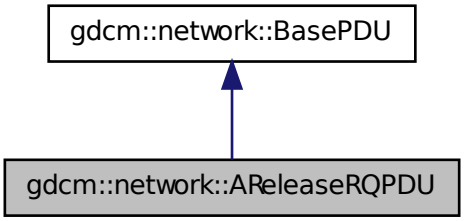
- gdcmAReleaseRPPDU.h

25.13 gdcm::network::AReleaseRQPDU Class Reference

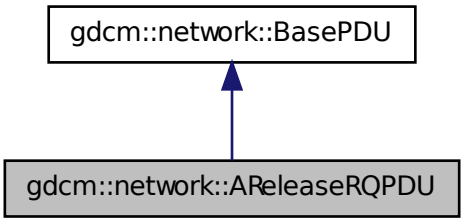
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

```
#include <gdcmAReleaseRQPDU.h>
```

Inheritance diagram for gdcm::network::AReleaseRQPDU:



Collaboration diagram for gdcm::network::AReleaseRQPDU:



Public Member Functions

- AReleaseRQPDU ()
- bool IsLastFragment () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.13.1 Detailed Description

AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

25.13.2 Constructor & Destructor Documentation

25.13.2.1 gdcm::network::AReleaseRQPDU::AReleaseRQPDU ( )

25.13.3 Member Function Documentation

25.13.3.1 bool gdcm::network::AReleaseRQPDU::IsLastFragment ( ) const [inline],[virtual]

Implements gdcm::network::BasePDU.

25.13.3.2 void gdcm::network::AReleaseRQPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

25.13.3.3 std::istream& gdcm::network::AReleaseRQPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

25.13.3.4 size_t gdcm::network::AReleaseRQPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

25.13.3.5 const std::ostream& gdcm::network::AReleaseRQPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

- gdcmAReleaseRQPDU.h

25.14 gdcm::network::ARTIMTimer Class Reference

ARTIMTimer This file contains the code for the ARTIM timer.

```
#include <gdcmARTIMTimer.h>
```

Public Member Functions

- ARTIMTimer ()
- double GetElapsedTime () const
- bool GetHasExpired () const
- double GetTimeout () const
- void SetTimeout (double inTimeout)
- void Start ()
- void Stop ()

25.14.1 Detailed Description

ARTIMTimer This file contains the code for the ARTIM timer.

Basically, the ARTIM timer will just get the wall time when it's started, and then can be queried for the current time, and then can be stopped (ie, the start time reset).

Because we're trying to do this without threading, we should be able to 'start' the ARTIM timer by this mechanism, and then when waiting for a particular response, tight loop that with sleep calls and determinations of when the ARTIM timer has reached its peak. As such, this isn't a strict 'timer' in the traditional sense of the word, but more of a time keeper.

There can be only one ARTIM timer per connection.

25.14.2 Constructor & Destructor Documentation

25.14.2.1 gdcm::network::ARTIMTimer::ARTIMTimer ( )

25.14.3 Member Function Documentation

25.14.3.1 double gdcm::network::ARTIMTimer::GetElapsedTime ( ) const

25.14.3.2 bool gdcm::network::ARTIMTimer::GetHasExpired ( ) const

25.14.3.3 double gdcm::network::ARTIMTimer::GetTimeout ( ) const

25.14.3.4 void gdcm::network::ARTIMTimer::SetTimeout (double *inTimeout*)

25.14.3.5 void gdcm::network::ARTIMTimer::Start ( )

25.14.3.6 void gdcm::network::ARTIMTimer::Stop ( )

The documentation for this class was generated from the following file:

- gdcmARTIMTimer.h

25.15 gdcm::ASN1 Class Reference

Class for ASN1.

```
#include <gdcmASN1.h>
```

Public Member Functions

- ASN1 ()
- ∼ASN1 ()

Static Public Member Functions

- static bool ParseDump (const char ∗array, size_t length)
- static bool ParseDumpFile (const char ∗filename)

Protected Member Functions

- int TestPBKDF2 ()

25.15.1 Detailed Description

Class for ASN1.

25.15.2 Constructor & Destructor Documentation

25.15.2.1 gdcm::ASN1::ASN1 ( )

25.15.2.2 gdcm::ASN1::∼ASN1 ( )

25.15.3 Member Function Documentation

25.15.3.1 static bool gdcm::ASN1::ParseDump (const char ∗ *array*, size_t *length*) [static]

25.15.3.2 static bool gdcm::ASN1::ParseDumpFile (const char ∗ *filename*) [static]

25.15.3.3 int gdcm::ASN1::TestPBKDF2 ( ) [protected]

The documentation for this class was generated from the following file:

- gdcmASN1.h

25.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference

AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#include <gdcmAsynchronousOperationsWindowSub.h>
```

Public Member Functions

- AsynchronousOperationsWindowSub ()
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.16.1 Detailed Description

AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

25.16.2 Constructor & Destructor Documentation

25.16.2.1 gdcm::network::AsynchronousOperationsWindowSub::AsynchronousOperationsWindowSub ( )

25.16.3 Member Function Documentation

25.16.3.1 std::istream& gdcm::network::AsynchronousOperationsWindowSub::Read (std::istream & *is*)

25.16.3.2 size_t gdcm::network::AsynchronousOperationsWindowSub::Size ( ) const

25.16.3.3 const std::ostream& gdcm::network::AsynchronousOperationsWindowSub::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

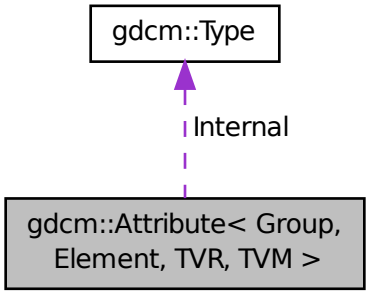
- gdcmAsynchronousOperationsWindowSub.h

25.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference

Attribute class This class use template metaprograming tricks to let the user know when the template instanciation does not match the public dictionary.

`#include <gdcmAttribute.h>`

Collaboration diagram for gdcm::Attribute< Group, Element, TVR, TVM >:



Public Types

- enum { VMType = VMToLength<TVM>::Length }
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))

- GDCM_STATIC_ASSERT (((VM::VMType) TVM &(VM::VMType)(TagToType< Group, Element >::VMType)))
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) TVM==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- ArrayType const & operator[] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

25.17.1 Detailed Description

template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>class gdcm::Attribute< Group, Element, TVR, TVM >

Attribute class This class use template metaprograming tricks to let the user know when the template instanciation does not match the public dictionary.

Typical example that compile is: Attribute<0x0008,0x9007> a = {"ORIGINAL","PRIMARY","T1","NONE"};

Examples that will NOT compile are:

Attribute<0x0018,0x1182, VR::IS, VM::VM1> fd1 = {}; // not enough parameters Attribute<0x0018,0x1182, VR::IS, VM::VM2> fd2 = {0,1,2}; // too many initializers Attribute<0x0018,0x1182, VR::IS, VM::VM3> fd3 = {0,1,2}; // VM3 is not valid Attribute<0x0018,0x1182, VR::UL, VM::VM2> fd3 = {0,1}; // UL is not valid VR

Examples:

CreateJPIPDataSet.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenFakeIdentifyFile.cxx, GetSequenceUltrasound.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndPrintAttributes.cxx, SortImage.cxx, StreamImageReaderTest.cxx, and VolumeSorter.cxx.

25.17.2 Member Typedef Documentation

25.17.2.1 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType

25.17.3 Member Enumeration Documentation

25.17.3.1 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> anonymous enum

Enumerator:

VMType

25.17.4 Member Function Documentation

25.17.4.1 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))

25.17.4.2 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (((VM::VMType) TVM &(VM::VMType)(TagToType< Group, Element >::VMType)))

25.17.4.3 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) TVM==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))

25.17.4.4 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::Internal, gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VRASCII.

25.17.4.5 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM ( ) [inline],[static]

25.17.4.6 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR ( ) [inline], [static]

25.17.4.7 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues ( ) const [inline]

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue(), gdcm::Attribute< Group, Element, TVR, TVM >::operator!=(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=(), gdcm::Attribute< Group, Element, TVR, TVM >::operator<(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator<(), gdcm::Attribute< Group, Element, TVR, TVM >::operator==(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator==(), gdcm::Attribute< Group, Element, TVR, TVM >::Print(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, TVM >::SetValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetValues(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues().

25.17.4.8 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag ( ) [inline], [static]

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::Print(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcm::Attribute< Group, Element, TVR, TVM >::Set(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet().

25.17.4.9 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> ArrayType& gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (unsigned int *idx* = 0) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::operator[](), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[]().

25.17.4.10 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> ArrayType const& gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (unsigned int *idx* = 0) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.17.4.11 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> const ArrayType* gdcm::Attribute< Group, Element, TVR, TVM >::GetValues ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::operator!=(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=(), gdcm::Attribute< Group, Element, TVR, TVM >::operator<(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator<(), gdcm::Attribute< Group, Element, TVR, TVM >::operator==(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator==().

25.17.4.12 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static VM gdcm::Attribute< Group, Element, TVR, TVM >::GetVM ( ) [inline], [static]

Referenced by gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print().

25.17.4.13 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR ( ) [inline], [static]

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

25.17.4.14 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (const Attribute< Group, Element, TVR, TVM > & *att*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.17.4.15 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (const Attribute< Group, Element, TVR, TVM > & *att*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.17.4.16 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (const Attribute< Group, Element, TVR, TVM > & *att*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.17.4.17 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> ArrayType& gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (unsigned int *idx*) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValue().

25.17.4.18 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> ArrayType const& gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (unsigned int *idx*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValue().

25.17.4.19 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::Print (std::ostream & *os*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.17.4.20 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::Set (DataSet const & *ds*) [inline]

References gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

25.17.4.21 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (const ByteValue * *bv*) [inline], [protected]

References gdcm::ByteValue::GetLength(), gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::ByteValue::GetPointer(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

25.17.4.22 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (const ByteValue * *bv*) [inline], [protected]

References gdcm::ByteValue::GetLength(), gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::ByteValue::GetPointer(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement().

25.17.4.23 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (DataElement const & *de*) [inline]

References gdcm::DataElement::GetByteValue(), gdcm::Tag::GetGroup(), gdcm::DataElement::GetTag(), gdcm::-Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::VR::INVALID, gdcm::DataElement::IsEmpty(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), and gdcm::VR::UN.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::Set(), gdcm::Attribute< Group, Element, TVR, VM::-VM1 >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet().

25.17.4.24 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (DataSet const & *ds*) [inline]

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::IsEmpty(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFrom-DataElement().

25.17.4.25 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (ArrayType *v*, unsigned int *idx* = 0) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.17.4.26 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (const ArrayType * *array*, unsigned int *numel* = VMType) [inline]

Examples:

LargeVRDSExplicit.cxx.

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

Referenced by gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues().

25.17.5 Member Data Documentation

25.17.5.1 template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]

Referenced by gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute(), gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm-::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, T-VM >::GetValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), gdcm::Attribute<

Group, Element, TVR, VM::VM1 >::GetValues(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues(), gdcm::Attribute< Group, Element, TVR, TVM >::operator!=(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=(), gdcm::Attribute< Group, Element, TVR, TVM >::operator<(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator<(), gdcm::Attribute< Group, Element, TVR, TVM >::operator==(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator==(), gdcm::Attribute< Group, Element, TVR, TVM >::Print(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, TVM >::SetValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetValues(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::~Attribute().

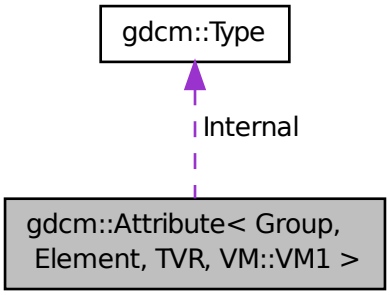
The documentation for this class was generated from the following file:

- gdcmAttribute.h

25.18 gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1 >:



Public Types

- enum { VMType = VMToLength<VM::VM1>::Length }
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (VMToLength< VM::VM1 >::Length==1)
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))

- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue ()
- ArrayType const & GetValue () const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

25.18.1 Member Typedef Documentation

25.18.1.1 template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1 >::ArrayType

25.18.2 Member Enumeration Documentation

25.18.2.1 template<uint16_t Group, uint16_t Element, int TVR> anonymous enum

Enumerator:

VMType

25.18.3 Member Function Documentation

25.18.3.1 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (VMToLength< VM::VM1 >::Length == 1)

25.18.3.2 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))

25.18.3.3 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))

25.18.3.4 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))

25.18.3.5 template<uint16_t Group, uint16_t Element, int TVR> DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::Internal, gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VRASCII.

25.18.3.6 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVM ( ) [inline], [static]

25.18.3.7 template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR ( ) [inline], [static]

25.18.3.8 template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues ( ) const [inline]

25.18.3.9 template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag ( ) [inline], [static]

25.18.3.10 template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( ) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.11 template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.12 template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.13 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVM ( ) [inline], [static]

References gdcm::VM::VM1.

25.18.3.14 template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR ( ) [inline], [static]

25.18.3.15 template<uint16_t Group, uint16_t Element, int TVR> bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!= (const Attribute< Group, Element, TVR, VM::VM1 > & *att*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.16 template<uint16_t Group, uint16_t Element, int TVR> bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator< (const Attribute< Group, Element, TVR, VM::VM1 > & *att*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.17 template<uint16_t Group, uint16_t Element, int TVR> bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator== (const Attribute< Group, Element, TVR, VM::VM1 > & *att*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.18 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print (std::ostream & *os*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.19 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set (DataSet const & *ds*) [inline]

References gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

25.18.3.20 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue (const ByteValue * *bv*) [inline], [protected]

References gdcm::ByteValue::GetLength(), gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::ByteValue::GetPointer(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.21 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap (const ByteValue * *bv*) [inline], [protected]

References gdcm::ByteValue::GetLength(), gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::ByteValue::GetPointer(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.22 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement (DataElement const & *de*) [inline]

References gdcm::DataElement::GetByteValue(), gdcm::Tag::GetGroup(), gdcm::DataElement::GetTag(), gdcm::-Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::VR::INVALID, gdcm::DataElement::IsEmpty(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), and gdcm::VR::UN.

25.18.3.23 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet (DataSet const & *ds*) [inline]

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::IsEmpty(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFrom-DataElement().

25.18.3.24 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue (ArrayType *v*) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.4 Member Data Documentation

25.18.4.1 template<uint16_t Group, uint16_t Element, int TVR> ArrayType gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Internal

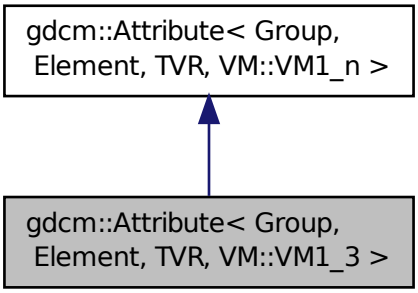
The documentation for this class was generated from the following file:

- gdcmAttribute.h

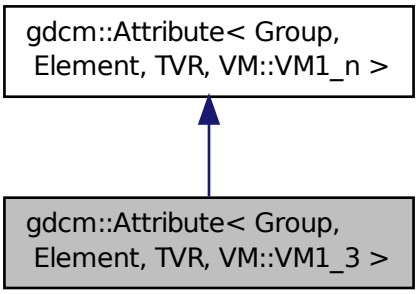
25.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >:



Public Member Functions

- VM GetVM () const

Additional Inherited Members

25.19.1 Member Function Documentation

25.19.1.1 template<uint16_t Group, uint16_t Element, int TVR> VM gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >::GetVM ( ) const [inline]

References gdcm::VM::VM1_3.

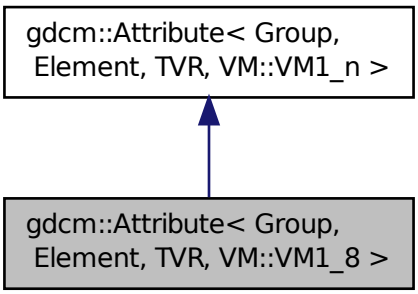
The documentation for this class was generated from the following file:

- gdcmAttribute.h

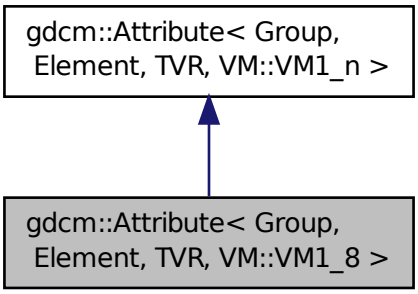
25.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >:



Public Member Functions

- VM GetVM () const

Additional Inherited Members

25.20.1 Member Function Documentation

25.20.1.1 template<uint16_t Group, uint16_t Element, int TVR> VM gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM ( ) const [inline]

References gdcm::VM::VM1_8.

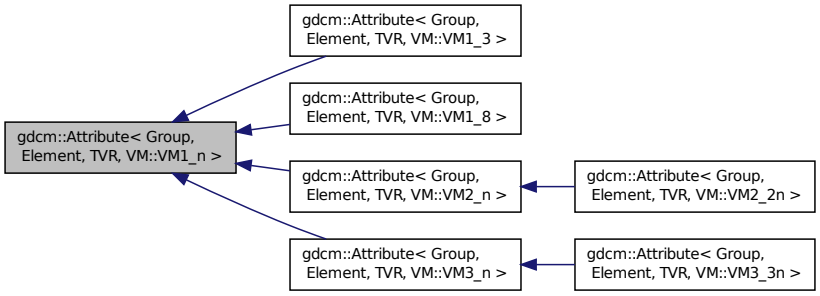
The documentation for this class was generated from the following file:

- gdcmAttribute.h

25.21 gdcm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference

`#include <gdcmAttribute.h>`

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_n >:



Public Types

- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- Attribute ()
- ~Attribute ()
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const

- ArrayType & operator[] (unsigned int idx)
- ArrayType const & operator[] (unsigned int idx) const
- void Print (std::ostream &os) const
- void SetFromDataElement (DataElement const &de)
- void SetNumberOfValues (unsigned int numel)
- void SetValue (unsigned int idx, ArrayType v)
- void SetValue (ArrayType v)
- void SetValues (const ArrayType *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Protected Member Functions

- void SetByteValue (const ByteValue *bv)

25.21.1 Member Typedef Documentation

25.21.1.1 template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::ArrayType

25.21.2 Constructor & Destructor Documentation

25.21.2.1 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute ( ) [inline], [explicit]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.21.2.2 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::~Attribute ( ) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.21.3 Member Function Documentation

25.21.3.1 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))

25.21.3.2 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))

25.21.3.3 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))

25.21.3.4 template<uint16_t Group, uint16_t Element, int TVR> DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::Internal, gdcm::DataElement::SetByteValue(), gdcm::DataElement::-SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VRASCII.

25.21.3.5 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM ( ) [inline], [static]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetVM().

25.21.3.6 template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR ( ) [inline], [static]

25.21.3.7 template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues ( ) const [inline]

25.21.3.8 template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag ( ) [inline], [static]

25.21.3.9 template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.21.3.10 template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.21.3.11 template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.21.3.12 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM ( ) [inline], [static]

Reimplemented in gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, and gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >.

References gdcm::VM::VM1_n.

25.21.3.13 template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR ( ) [inline], [static]

25.21.3.14 template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (unsigned int *idx*) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValue().

25.21.3.15 template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (unsigned int *idx*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValue().

25.21.3.16 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print (std::ostream & *os*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVM(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.21.3.17 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue (const ByteValue * *bv*) [inline], [protected]

References gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetValues().

25.21.3.18 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement (DataElement const & *de*) [inline]

References gdcm::DataElement::GetByteValue(), gdcm::Tag::GetGroup(), gdcm::DataElement::GetTag(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::DataElement::IsEmpty(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue().

25.21.3.19 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues (unsigned int *numel*) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::SetValues().

25.21.3.20 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (unsigned int *idx*, ArrayType *v*) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.21.3.21 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (ArrayType *v*) [inline]

References SetValue().

Referenced by SetValue().

25.21.3.22 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues (const ArrayType * *array*, unsigned int *numel*, bool *own* = `false`) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

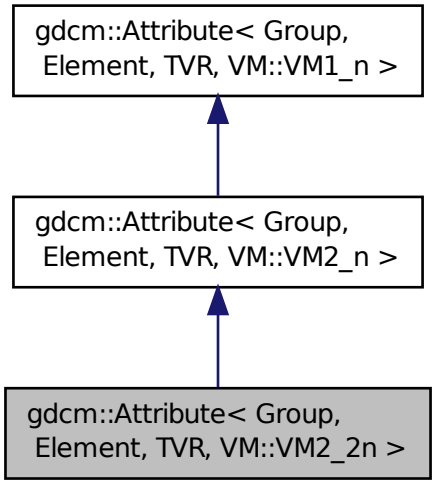
The documentation for this class was generated from the following file:

- gdcmAttribute.h

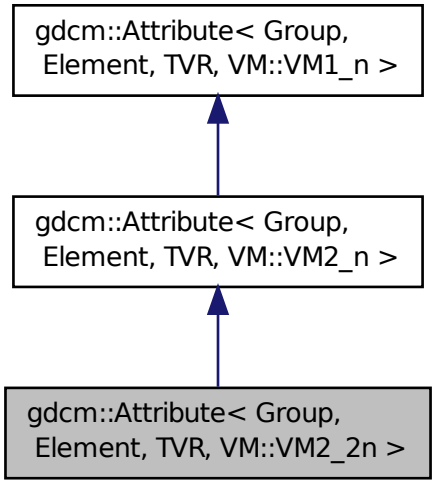
25.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference

`#include <gdcmAttribute.h>`

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >:



Static Public Member Functions

- static VM GetVM ()

Additional Inherited Members

25.22.1 Member Function Documentation

25.22.1.1 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM ( ) [inline], [static]

Reimplemented from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >.

References gdcm::VM::VM2_2n.

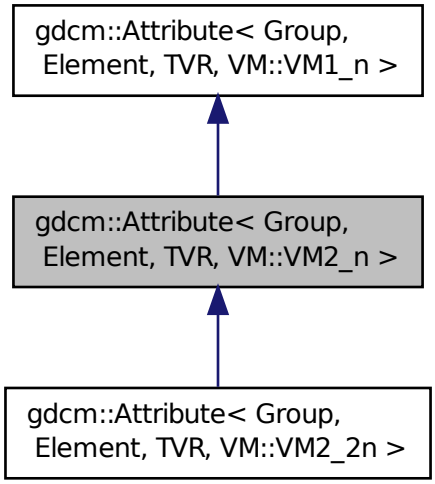
The documentation for this class was generated from the following file:

- gdcmAttribute.h

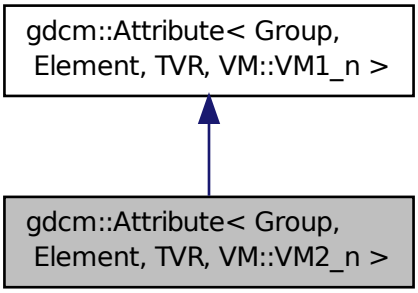
25.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_n >:



Public Member Functions

- VM GetVM () const

Additional Inherited Members

25.23.1 Member Function Documentation

25.23.1.1 template<uint16_t Group, uint16_t Element, int TVR> VM gdcm::Attribute< Group, Element, TVR, VM::VM2_n >::GetVM ( ) const [inline]

References gdcm::VM::VM2_n.

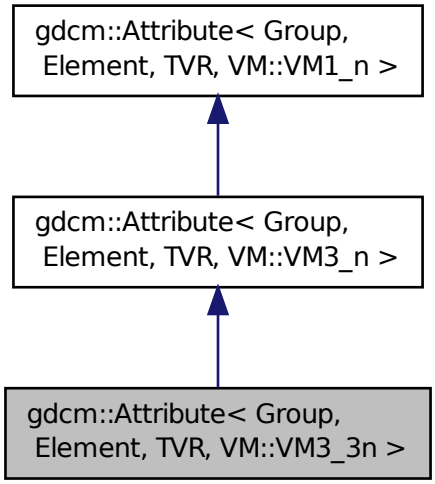
The documentation for this class was generated from the following file:

- gdcmAttribute.h

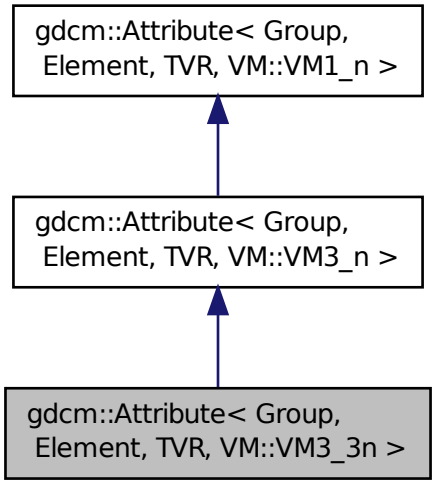
25.24 gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >:



Static Public Member Functions

- static VM GetVM ()

25.24.1 Member Function Documentation

25.24.1.1 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM ( ) [inline], [static]

Reimplemented from gdcm::Attribute< Group, Element, TVR, VM::VM3_n >.

References gdcm::VM::VM3_3n.

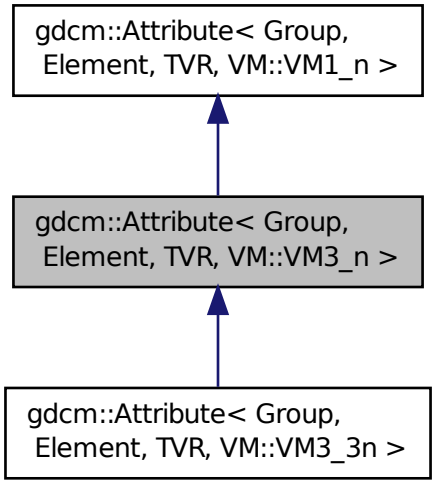
The documentation for this class was generated from the following file:

- gdcmAttribute.h

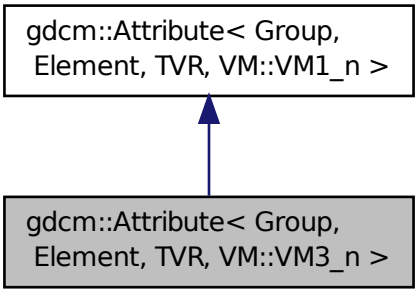
25.25 gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_n >:



Static Public Member Functions

- static VM GetVM ()

Additional Inherited Members

25.25.1 Member Function Documentation

25.25.1.1 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM3_n >::GetVM ( ) [inline], [static]

Reimplemented from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >.

Reimplemented in gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >.

References gdcm::VM::VM3_n.

The documentation for this class was generated from the following file:

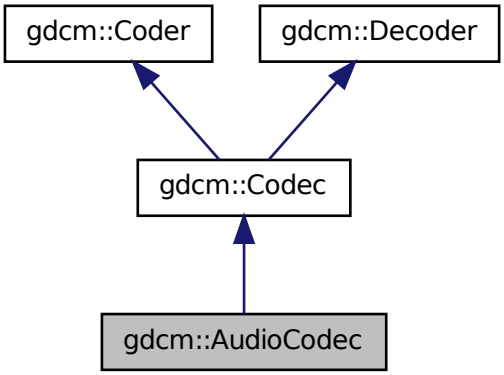
- gdcmAttribute.h

25.26 gdcm::AudioCodec Class Reference

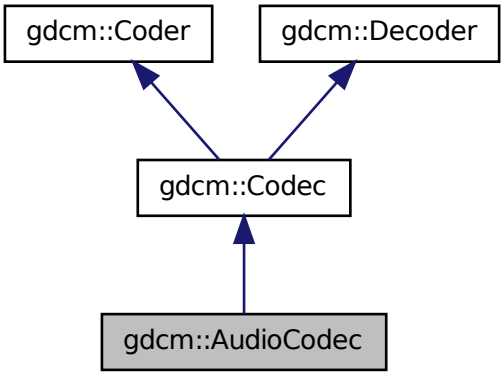
AudioCodec.

`#include <gdcmAudioCodec.h>`

Inheritance diagram for gdcm::AudioCodec:



Collaboration diagram for gdcm::AudioCodec:



Public Member Functions

- AudioCodec ()
- ∼AudioCodec ()
- bool CanCode (TransferSyntax const &) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

25.26.1 Detailed Description

AudioCodec.

25.26.2 Constructor & Destructor Documentation

25.26.2.1 gdcm::AudioCodec::AudioCodec ( )

25.26.2.2 gdcm::AudioCodec::∼AudioCodec ( )

25.26.3 Member Function Documentation

25.26.3.1 bool gdcm::AudioCodec::CanCode (TransferSyntax const &) const [inline], [virtual]

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

25.26.3.2 bool gdcm::AudioCodec::CanDecode (TransferSyntax const &) const [inline],[virtual]

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

25.26.3.3 bool gdcm::AudioCodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::Decoder.

The documentation for this class was generated from the following file:

- gdcmAudioCodec.h

25.27 gdcm::Base64 Class Reference

Class for Base64.

`#include <gdcmBase64.h>`

Public Member Functions

- Base64 ()
- ~Base64 ()

Static Public Member Functions

- static int Decode (char *dst, int dlen, const char *src, int slen)

  Decode a base64-formatted buffer.
- static int Encode (char *dst, int dlen, const char *src, int slen)

  Encode a buffer into base64 format.
- static int GetDecodeLength (const char *src, int slen)
- static int GetEncodeLength (const char *src, int slen)

25.27.1 Detailed Description

Class for Base64.

25.27.2 Constructor & Destructor Documentation

25.27.2.1 gdcm::Base64::Base64 ( )

25.27.2.2 gdcm::Base64::~Base64 ( )

25.27.3 Member Function Documentation

25.27.3.1 static int gdcm::Base64::Decode (char * *dst*, int *dlen*, const char * *src*, int *slen*) [static]

Decode a base64-formatted buffer.

Parameters

dst	destination buffer
dlen	size of the buffer
src	source buffer
slen	amount of data to be decoded

Returns

0 if successful

25.27.3.2 static int gdcm::Base64::Encode (char * *dst*, int *dlen*, const char * *src*, int *slen*) [static]

Encode a buffer into base64 format.

Parameters

dst	destination buffer
dlen	size of the buffer
src	source buffer
slen	amount of data to be encoded

Returns

0 if successful

25.27.3.3 static int gdcm::Base64::GetDecodeLength (const char * *src*, int *slen*) [static]

Call this function with *dlen = 0 to obtain the required buffer size in *dlen

25.27.3.4 static int gdcm::Base64::GetEncodeLength (const char * *src*, int *slen*) [static]

Call this function with dlen = 0 to obtain the required buffer size in dlen

The documentation for this class was generated from the following file:

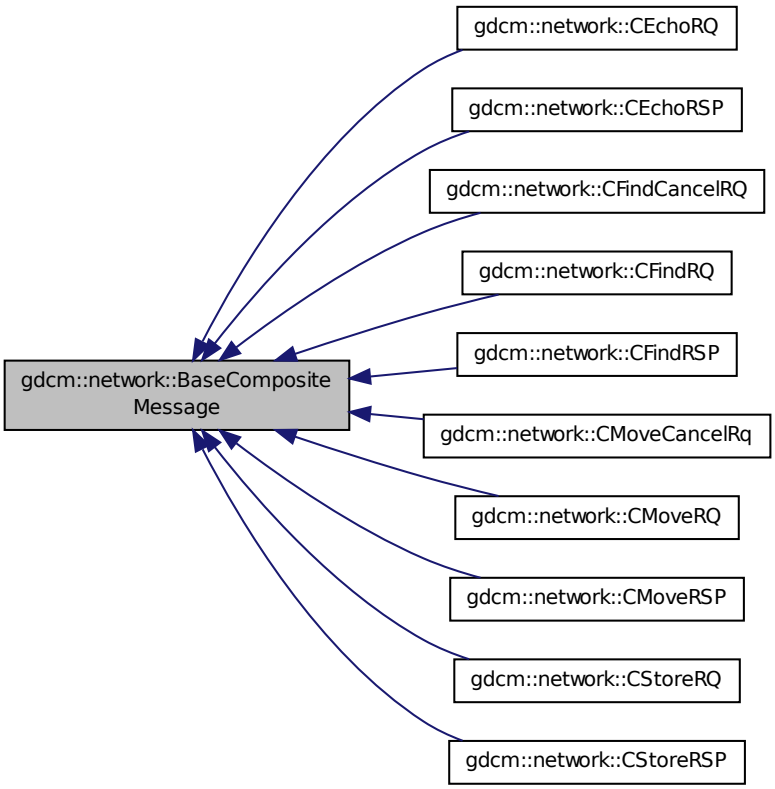
- gdcmBase64.h

25.28 gdcm::network::BaseCompositeMessage Class Reference

BaseCompositeMessage The Composite events described in section 3.7-2009 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2009 of the standard, and then fill in appropriate values in their datasets.

```
#include <gdcmBaseCompositeMessage.h>
```

Inheritance diagram for gdcm::network::BaseCompositeMessage:



Public Member Functions

- virtual std::vector
  < PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0

25.28.1 Detailed Description

BaseCompositeMessage The Composite events described in section 3.7-2009 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2009 of the standard, and then fill in appropriate values in their datasets.

So, for the five composites:

- C-ECHO
- C-FIND
- C-MOVE
- C-GET

- C-STORE there are a series of messages. However, all of these messages are obtained as part of a PDataPDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, gdcmCompositeP-DUFactory.

This is an abstract class. It cannot be instantiated on its own.

25.28.2 Member Function Documentation

25.28.2.1 virtual std::vector<PresentationDataValue> gdcm::network::BaseCompositeMessage::ConstructPDV (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [pure virtual]

Implemented in gdcm::network::CMoveRQ, gdcm::network::CFindRQ, and gdcm::network::CEchoRQ.

The documentation for this class was generated from the following file:

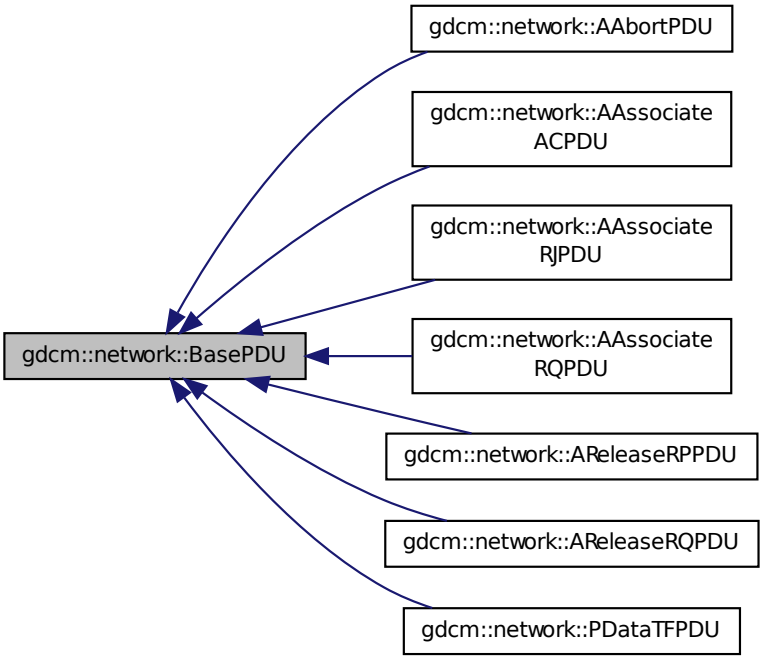
- gdcmBaseCompositeMessage.h

25.29 gdcm::network::BasePDU Class Reference

BasePDU base class for PDUs.

```
#include <gdcmBasePDU.h>
```

Inheritance diagram for gdcm::network::BasePDU:



Public Member Functions

- virtual ~BasePDU ()
- virtual bool IsLastFragment () const =0
- virtual void Print (std::ostream &os) const =0
- virtual std::istream & Read (std::istream &is)=0
- virtual size_t Size () const =0
- virtual const std::ostream & Write (std::ostream &os) const =0

25.29.1 Detailed Description

BasePDU base class for PDUs.

all PDUs start with the first ten bytes as specified: 01 PDU type 02 reserved 3-6 PDU Length (unsigned) 7-10 variable

on some, 7-10 are split (7-8 as protocol version in Associate-RQ, for instance, while associate-rj splits those four bytes differently).

Also common to all the PDUs is their ability to read and write to a stream.

So, let's just get them all bunched together into one (abstract) class, shall we?

Why? 1) so that the ULEvent can have the PDU stored in it, since the event takes PDUs and not other class structures (other class structures get converted into PDUs) 2) to make reading PDUs in the event loop cleaner

but! leave Mathieu's original classes untouched at this point (except for inheriting from this class) because those work, at least with c-echo.

25.29.2 Constructor & Destructor Documentation

25.29.2.1 virtual gdcm::network::BasePDU::~BasePDU ( ) [inline],[virtual]

25.29.3 Member Function Documentation

25.29.3.1 virtual bool gdcm::network::BasePDU::IsLastFragment ( ) const [pure virtual]

Implemented in gdcm::network::AAssociateRQPDU, gdcm::network::AAssociateACPDU, gdcm::network::PDataTFPDU, gdcm::network::AAbortPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, and gdcm::network::AReleaseRQPDU.

25.29.3.2 virtual void gdcm::network::BasePDU::Print (std::ostream & *os*) const [pure virtual]

Implemented in gdcm::network::AAssociateRQPDU, gdcm::network::AAssociateACPDU, gdcm::network::PDataTFPDU, gdcm::network::AAbortPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::AAssociateRJPDU.

25.29.3.3 virtual std::istream& gdcm::network::BasePDU::Read (std::istream & *is*) [pure virtual]

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTFPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::AAbortPDU.

25.29.3.4 virtual size_t gdcm::network::BasePDU::Size ( ) const [pure virtual]

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTFPDU, gdcm::network::AAbortPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, and gdcm::network::AReleaseRQPDU.

25.29.3.5 virtual const std::ostream& gdcm::network::BasePDU::Write (std::ostream & *os*) const [pure virtual]

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTFPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::AAbortPDU.

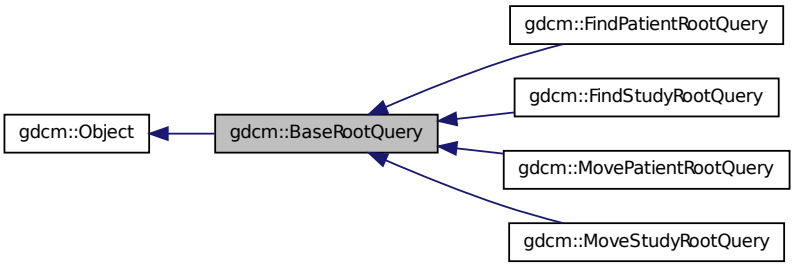
The documentation for this class was generated from the following file:

- gdcmBasePDU.h

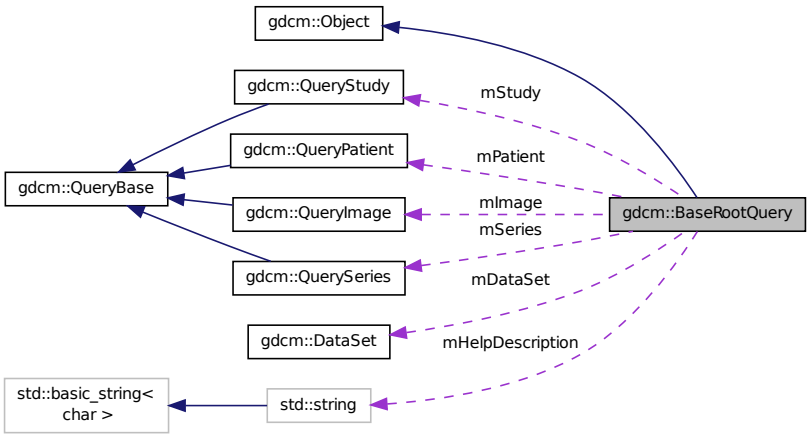
25.30 gdcm::BaseRootQuery Class Reference

```
#include <gdcmBaseRootQuery.h>
```

Inheritance diagram for gdcm::BaseRootQuery:



Collaboration diagram for gdcm::BaseRootQuery:



Public Member Functions

- virtual ~BaseRootQuery ()
- void AddQueryDataSet (const DataSet &ds)
- virtual UIDs::TSName GetAbstractSyntaxUID () const =0
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- DataSet & GetQueryDataSet ()
- virtual std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)=0
- virtual void InitializeDataSet (const EQueryLevel &inQueryLevel)=0
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- virtual bool ValidateQuery (bool inStrict=true) const =0
- virtual const std::ostream & WriteHelpFile (std::ostream &os)
- virtual bool WriteQuery (const std::string &inFileName)

Protected Member Functions

- BaseRootQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)

Protected Attributes

- DataSet mDataSet
- std::string mHelpDescription
- QueryImage mImage
- QueryPatient mPatient
- ERootType mRootType
- QuerySeries mSeries
- QueryStudy mStudy

Friends

- class QueryFactory

25.30.1 Constructor & Destructor Documentation

25.30.1.1 gdcm::BaseRootQuery::BaseRootQuery ( ) [protected]

25.30.1.2 virtual gdcm::BaseRootQuery::~BaseRootQuery ( ) [virtual]

25.30.2 Member Function Documentation

25.30.2.1 void gdcm::BaseRootQuery::AddQueryDataSet (const DataSet & *ds*)

25.30.2.2 virtual UIDs::TSName gdcm::BaseRootQuery::GetAbstractSyntaxUID ( ) const [pure virtual]

Implemented in gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::FindPatientRootQuery.

25.30.2.3 DataSet const& gdcm::BaseRootQuery::GetQueryDataSet ( ) const

Set/Get the internal representation of the query as a DataSet.

25.30.2.4 DataSet& gdcm::BaseRootQuery::GetQueryDataSet ( )

25.30.2.5 virtual std::vector<Tag> gdcm::BaseRootQuery::GetTagListByLevel (const EQueryLevel & *inQueryLevel*) [pure virtual]

this function will return all tags at a given query level, so that ∗they maybe selected for searching. The boolean forFind is true ∗if the query is a find query, or false for a move query.

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, and gdcm::MoveStudyRootQuery.

25.30.2.6 virtual void gdcm::BaseRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [pure virtual]

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, and gdcm::MoveStudyRootQuery.

25.30.2.7 void gdcm::BaseRootQuery::SetSearchParameter (const Tag & *inTag,* const DictEntry & *inDictEntry,* const std::string & *inValue*) [protected]

25.30.2.8 void gdcm::BaseRootQuery::SetSearchParameter (const Tag & *inTag,* const std::string & *inValue*)

25.30.2.9 void gdcm::BaseRootQuery::SetSearchParameter (const std::string & *inKeyword,* const std::string & *inValue*)

25.30.2.10 virtual bool gdcm::BaseRootQuery::ValidateQuery (bool *inStrict* = true) const [pure virtual]

have to be able to ensure that ∗0x8,0x52 is set (which will be true if InitializeDataSet is called...) ∗that the level is appropriate (ie, not setting PATIENT for a study query ∗that the tags in the query match the right level (either required, unique, optional) ∗by default, this function checks to see if the query is for finding, which is more ∗permissive than for moving. For moving, only the unique tags are allowed. ∗10 Jan 2011: adding in the 'strict' mode. ∗according to the standard (at least, how I've read it), only tags for a particular ∗level should be allowed in a particular query (ie, just series level tags in a series ∗level query). However, it seems that dcm4chee doesn't share that interpretation. ∗So, if 'inStrict' is false, then tags from the current level and all higher levels ∗are now considered valid. So, if you're doing a non-strict series-level query, ∗tags from the patient and study level can be passed along as well.

Implemented in gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::FindPatientRootQuery.

25.30.2.11 virtual const std::ostream& gdcm::BaseRootQuery::WriteHelpFile (std::ostream & *os*) [virtual]

25.30.2.12 virtual bool gdcm::BaseRootQuery::WriteQuery (const std::string & *inFileName*) [virtual]

25.30.3 Friends And Related Function Documentation

25.30.3.1 friend class QueryFactory [friend]

Reimplemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, and gdcm::MoveStudyRootQuery.

25.30.4 Member Data Documentation

25.30.4.1 DataSet gdcm::BaseRootQuery::mDataSet [protected]

25.30.4.2 std::string gdcm::BaseRootQuery::mHelpDescription [protected]

25.30.4.3 QueryImage gdcm::BaseRootQuery::mImage [protected]

25.30.4.4 QueryPatient gdcm::BaseRootQuery::mPatient [protected]

25.30.4.5 ERootType gdcm::BaseRootQuery::mRootType [protected]

25.30.4.6 QuerySeries gdcm::BaseRootQuery::mSeries [protected]

25.30.4.7 QueryStudy gdcm::BaseRootQuery::mStudy [protected]

The documentation for this class was generated from the following file:

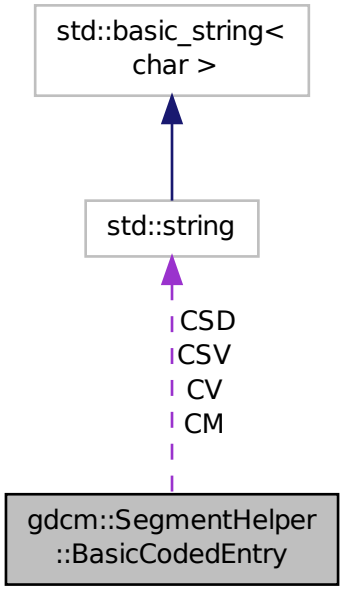
- gdcmBaseRootQuery.h

25.31 gdcm::SegmentHelper::BasicCodedEntry Struct Reference

This structure defines a basic coded entry with all of its attributes.

```
#include <gdcmSegmentHelper.h>
```

Collaboration diagram for gdcm::SegmentHelper::BasicCodedEntry:



Public Member Functions

- BasicCodedEntry ()

  Constructor.
- BasicCodedEntry (const char ∗a_CV, const char ∗a_CSD, const char ∗a_CM)

  constructor which defines type 1 attributes.
- BasicCodedEntry (const char ∗a_CV, const char ∗a_CSD, const char ∗a_CSV, const char ∗a_CM)

  constructor which defines attributes.
- bool IsEmpty (const bool checkOptionalAttributes=false) const

  Check if each attibutes of the basic coded entry is defined.

Public Attributes

- std::string CM

  Coding Scheme Version attribute.
- std::string CSD

  Code Value attribute.
- std::string CSV

  Coding Scheme Designator attribute.
- std::string CV

25.31.1 Detailed Description

This structure defines a basic coded entry with all of its attributes.

See also

PS 3.3 section 8.8.

25.31.2 Constructor & Destructor Documentation

25.31.2.1 gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ( ) [inline]

Constructor.

25.31.2.2 gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (const char * *a_CV*, const char * *a_CSD*, const char * *a_CM*) [inline]

constructor which defines type 1 attributes.

25.31.2.3 gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (const char * *a_CV*, const char * *a_CSD*, const char * *a_CSV*, const char * *a_CM*) [inline]

constructor which defines attributes.

25.31.3 Member Function Documentation

25.31.3.1 bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (const bool *checkOptionalAttributes* = false) const

Check if each attibutes of the basic coded entry is defined.

Parameters

checkOptional-Attributes	Check also type 1C attributes.

25.31.4 Member Data Documentation

25.31.4.1 std::string gdcm::SegmentHelper::BasicCodedEntry::CM

Coding Scheme Version attribute.

25.31.4.2 std::string gdcm::SegmentHelper::BasicCodedEntry::CSD

Code Value attribute.

25.31.4.3 std::string gdcm::SegmentHelper::BasicCodedEntry::CSV

Coding Scheme Designator attribute.

25.31.4.4 std::string gdcm::SegmentHelper::BasicCodedEntry::CV

The documentation for this struct was generated from the following file:

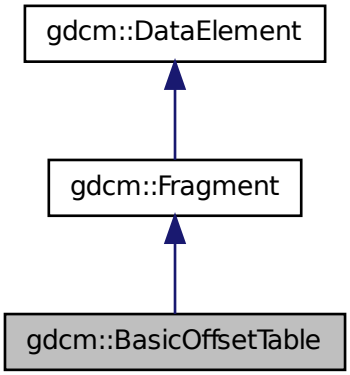
- gdcmSegmentHelper.h

25.32 gdcm::BasicOffsetTable Class Reference

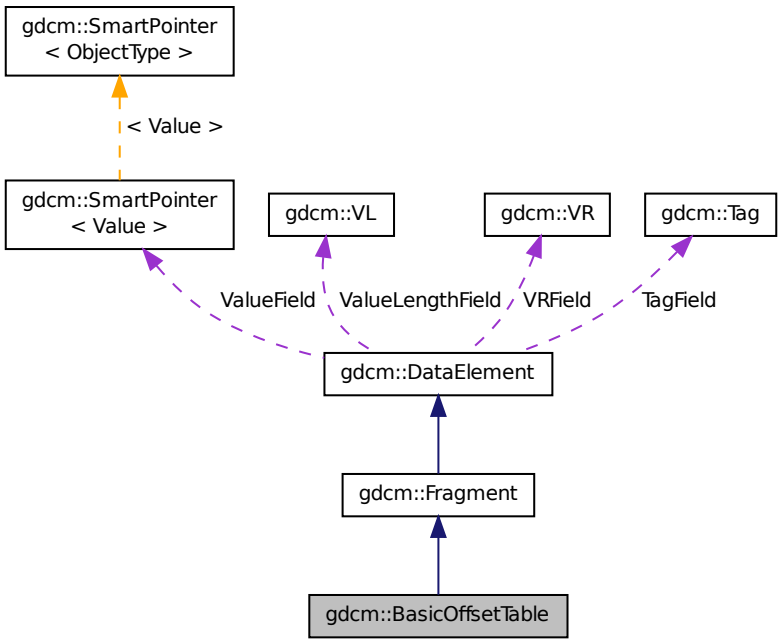
Class to represent a BasicOffsetTable.

`#include <gdcmBasicOffsetTable.h>`

Inheritance diagram for gdcm::BasicOffsetTable:



Collaboration diagram for gdcm::BasicOffsetTable:



Public Member Functions

- BasicOffsetTable ()
- template<typename TSwap >
  std::istream & Read (std::istream &is)

Friends

- std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)

25.32.1 Detailed Description

Class to represent a BasicOffsetTable.

25.32.2 Constructor & Destructor Documentation

25.32.2.1 gdcm::BasicOffsetTable::BasicOffsetTable ( ) [inline]

25.32.3 Member Function Documentation

25.32.3.1 template<typename TSwap > std::istream& gdcm::BasicOffsetTable::Read (std::istream & *is*) [inline]

Reimplemented from gdcm::Fragment.

25.32.4 Friends And Related Function Documentation

25.32.4.1 std::ostream& operator<< (std::ostream & *os,* const **BasicOffsetTable** & *val*) [friend]

The documentation for this class was generated from the following file:

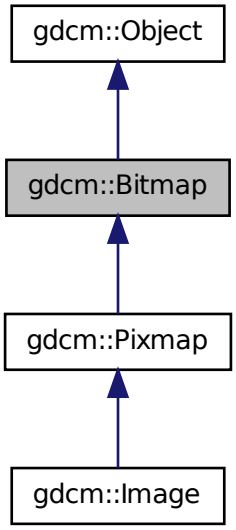
- gdcmBasicOffsetTable.h

25.33 gdcm::Bitmap Class Reference

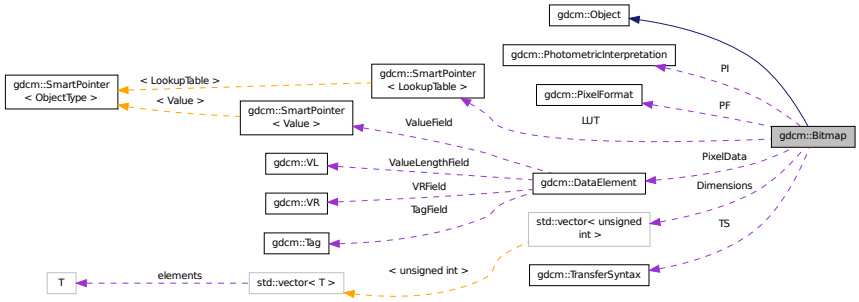
Bitmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

`#include <gdcmBitmap.h>`

Inheritance diagram for gdcm::Bitmap:



Collaboration diagram for gdcm::Bitmap:



Public Member Functions

- Bitmap ()
- ~Bitmap ()
- virtual bool AreOverlaysInPixelData () const
- void Clear ()
- bool GetBuffer (char *buffer) const

  Acces the raw data.
- unsigned long GetBufferLength () const
- unsigned int GetColumns () const
- const DataElement & GetDataElement () const
- DataElement & GetDataElement ()
- unsigned int GetDimension (unsigned int idx) const
- const unsigned int * GetDimensions () const

  Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...
- const LookupTable & GetLUT () const
- LookupTable & GetLUT ()
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const

  Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.
- const PhotometricInterpretation & GetPhotometricInterpretation () const

  return the photometric interpretation
- const PixelFormat & GetPixelFormat () const

  Get/Set PixelFormat.
- PixelFormat & GetPixelFormat ()
- unsigned int GetPlanarConfiguration () const

  return the planar configuration
- unsigned int GetRows () const
- const TransferSyntax & GetTransferSyntax () const
- bool IsEmpty () const
- bool IsLossy () const

  Return whether or not the image was compressed using a lossy compressor or not.
- bool IsTransferSyntaxCompatible (TransferSyntax const &ts) const
- void Print (std::ostream &) const

- void SetColumns (unsigned int col)
- void SetDataElement (DataElement const &de)
- void SetDimension (unsigned int idx, unsigned int dim)
- void SetDimensions (const unsigned int dims[3])
- void SetLossyFlag (bool f)

  Specifically set that the image was compressed using a lossy compression mechanism.
- void SetLUT (LookupTable const &lut)

  Set/Get LUT.
- void SetNeedByteSwap (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)
- void SetRows (unsigned int rows)
- void SetTransferSyntax (TransferSyntax const &ts)

  Transfer syntax.

Protected Types

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions

- bool ComputeLossyFlag ()
- bool GetBuffer2 (std::ostream &os) const
- bool TryJPEG2000Codec (char *buffer, bool &lossyflag) const
- bool TryJPEG2000Codec2 (std::ostream &os) const
- bool TryJPEGCodec (char *buffer, bool &lossyflag) const
- bool TryJPEGCodec2 (std::ostream &os) const
- bool TryJPEGLSCodec (char *buffer, bool &lossyflag) const
- bool TryKAKADUCodec (char *buffer, bool &lossyflag) const
- bool TryPVRGCodec (char *buffer, bool &lossyflag) const
- bool TryRAWCodec (char *buffer, bool &lossyflag) const
- bool TryRLECodec (char *buffer, bool &lossyflag) const

Protected Attributes

- std::vector< unsigned int > Dimensions
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- DataElement PixelData
- unsigned int PlanarConfiguration
- TransferSyntax TS

Friends

- class ImageChangeTransferSyntax
- class PixmapReader

25.33.1 Detailed Description

Bitmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

Examples:

ExtractIconFromFile.cxx.

25.33.2 Member Typedef Documentation

25.33.2.1 typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr [protected]

25.33.3 Constructor & Destructor Documentation

25.33.3.1 gdcm::Bitmap::Bitmap ( )

25.33.3.2 gdcm::Bitmap::~Bitmap ( )

25.33.4 Member Function Documentation

25.33.4.1 virtual bool gdcm::Bitmap::AreOverlaysInPixelData ( ) const [inline], [virtual]

Reimplemented in gdcm::Pixmap.

25.33.4.2 void gdcm::Bitmap::Clear ( )

25.33.4.3 bool gdcm::Bitmap::ComputeLossyFlag ( ) [protected]

25.33.4.4 bool gdcm::Bitmap::GetBuffer (char * *buffer*) const

Acces the raw data.

Examples:

ConvertToQImage.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

25.33.4.5 bool gdcm::Bitmap::GetBuffer2 (std::ostream & *os*) const [protected]

25.33.4.6 unsigned long gdcm::Bitmap::GetBufferLength ( ) const

Return the length of the image after decompression WARNING for palette color: It will NOT take into account the Palette Color thus you need to multiply this length by 3 if computing the size of equivalent RGB image

Examples:

ConvertToQImage.cxx, GenFakeImage.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

25.33.4.7 unsigned int gdcm::Bitmap::GetColumns ( ) const [inline]

25.33.4.8 const DataElement& gdcm::Bitmap::GetDataElement ( ) const [inline]

Examples:

ExtractIconFromFile.cxx.

25.33.4.9 DataElement& gdcm::Bitmap::GetDataElement ( ) [inline]

25.33.4.10 unsigned int gdcm::Bitmap::GetDimension (unsigned int *idx*) const

25.33.4.11 const unsigned int* gdcm::Bitmap::GetDimensions ( ) const

Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...

Examples:

ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, HelloVizWorld.cxx, and threadgdcm.cxx.

25.33.4.12 const LookupTable& gdcm::Bitmap::GetLUT ( ) const [inline]

Examples:

ExtractIconFromFile.cxx.

25.33.4.13 LookupTable& gdcm::Bitmap::GetLUT ( ) [inline]

25.33.4.14 bool gdcm::Bitmap::GetNeedByteSwap ( ) const [inline]

25.33.4.15 unsigned int gdcm::Bitmap::GetNumberOfDimensions ( ) const

Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.

Examples:

HelloVizWorld.cxx, and threadgdcm.cxx.

25.33.4.16 const PhotometricInterpretation& gdcm::Bitmap::GetPhotometricInterpretation ( ) const

return the photometric interpretation

Examples:

ConvertToQImage.cxx, ExtractIconFromFile.cxx, and HelloVizWorld.cxx.

25.33.4.17 const PixelFormat& gdcm::Bitmap::GetPixelFormat ( ) const [inline]

Get/Set PixelFormat.

Examples:

ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GetJPEGSamplePrecision.cxx, and threadgdcm.cxx.

25.33.4.18 PixelFormat& gdcm::Bitmap::GetPixelFormat ( ) [inline]

25.33.4.19 unsigned int gdcm::Bitmap::GetPlanarConfiguration ( ) const

return the planar configuration

25.33.4.20 unsigned int gdcm::Bitmap::GetRows ( ) const [inline]

25.33.4.21 const TransferSyntax& gdcm::Bitmap::GetTransferSyntax ( ) const [inline]

Examples:

ExtractIconFromFile.cxx.

25.33.4.22 bool gdcm::Bitmap::IsEmpty ( ) const [inline]

25.33.4.23 bool gdcm::Bitmap::IsLossy ( ) const

Return whether or not the image was compressed using a lossy compressor or not.

25.33.4.24 bool gdcm::Bitmap::IsTransferSyntaxCompatible (TransferSyntax const & *ts*) const

25.33.4.25 void gdcm::Bitmap::Print (std::ostream &) const [virtual]

Reimplemented from gdcm::Object.

Reimplemented in gdcm::Image, and gdcm::Pixmap.

Examples:

ExtractIconFromFile.cxx.

25.33.4.26 void gdcm::Bitmap::SetColumns (unsigned int *col*) [inline]

25.33.4.27 void gdcm::Bitmap::SetDataElement (DataElement const & *de*) [inline]

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, and iU22tomultisc.cxx.

25.33.4.28 void gdcm::Bitmap::SetDimension (unsigned int *idx,* unsigned int *dim*)

Examples:

csa2img.cxx, GenFakeImage.cxx, and iU22tomultisc.cxx.

25.33.4.29 void gdcm::Bitmap::SetDimensions (const unsigned int *dims[3]*)

Examples:

CreateARGBImage.cxx, and CreateCMYKImage.cxx.

25.33.4.30 void gdcm::Bitmap::SetLossyFlag (bool *f*) [inline]

Specifically set that the image was compressed using a lossy compression mechanism.

25.33.4.31 void gdcm::Bitmap::SetLUT (**LookupTable** const & *lut*) [inline]

Set/Get LUT.

25.33.4.32 void gdcm::Bitmap::SetNeedByteSwap (bool *b*) [inline]

25.33.4.33 void gdcm::Bitmap::SetNumberOfDimensions (unsigned int *dim*)

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

25.33.4.34 void gdcm::Bitmap::SetPhotometricInterpretation (**PhotometricInterpretation** const & *pi*)

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, and iU22tomultisc.cxx.

25.33.4.35 void gdcm::Bitmap::SetPixelFormat (**PixelFormat** const & *pf*) [inline]

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

References gdcm::PixelFormat::Validate().

25.33.4.36 void gdcm::Bitmap::SetPlanarConfiguration (unsigned int *pc*)

Warning

you need to call SetPixelFormat first (before SetPlanarConfiguration) for consistency checking

25.33.4.37 void gdcm::Bitmap::SetRows (unsigned int *rows*) [inline]

25.33.4.38 void gdcm::Bitmap::SetTransferSyntax (TransferSyntax const & *ts*) [inline]

Transfer syntax.

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, and MergeTwoFiles.cxx.

25.33.4.39 bool gdcm::Bitmap::TryJPEG2000Codec (char * *buffer,* bool & *lossyflag*) const [protected]

25.33.4.40 bool gdcm::Bitmap::TryJPEG2000Codec2 (std::ostream & *os*) const [protected]

25.33.4.41 bool gdcm::Bitmap::TryJPEGCodec (char * *buffer,* bool & *lossyflag*) const [protected]

25.33.4.42 bool gdcm::Bitmap::TryJPEGCodec2 (std::ostream & *os*) const [protected]

25.33.4.43 bool gdcm::Bitmap::TryJPEGLSCodec (char * *buffer,* bool & *lossyflag*) const [protected]

25.33.4.44 bool gdcm::Bitmap::TryKAKADUCodec (char * *buffer,* bool & *lossyflag*) const [protected]

25.33.4.45 bool gdcm::Bitmap::TryPVRGCodec (char * *buffer,* bool & *lossyflag*) const [protected]

25.33.4.46 bool gdcm::Bitmap::TryRAWCodec (char * *buffer,* bool & *lossyflag*) const [protected]

25.33.4.47 bool gdcm::Bitmap::TryRLECodec (char * *buffer,* bool & *lossyflag*) const [protected]

25.33.5 Friends And Related Function Documentation

25.33.5.1 friend class ImageChangeTransferSyntax [friend]

25.33.5.2 friend class PixmapReader [friend]

25.33.6 Member Data Documentation

25.33.6.1 std::vector<unsigned int> gdcm::Bitmap::Dimensions [protected]

25.33.6.2 bool gdcm::Bitmap::LossyFlag [protected]

25.33.6.3 LUTPtr gdcm::Bitmap::LUT [protected]

25.33.6.4 bool gdcm::Bitmap::NeedByteSwap [protected]

25.33.6.5 unsigned int gdcm::Bitmap::NumberOfDimensions [protected]

25.33.6.6 PixelFormat gdcm::Bitmap::PF [protected]

25.33.6.7 PhotometricInterpretation gdcm::Bitmap::PI [protected]

25.33.6.8 DataElement gdcm::Bitmap::PixelData [protected]

25.33.6.9 unsigned int gdcm::Bitmap::PlanarConfiguration [protected]

25.33.6.10 TransferSyntax gdcm::Bitmap::TS [protected]

The documentation for this class was generated from the following file:

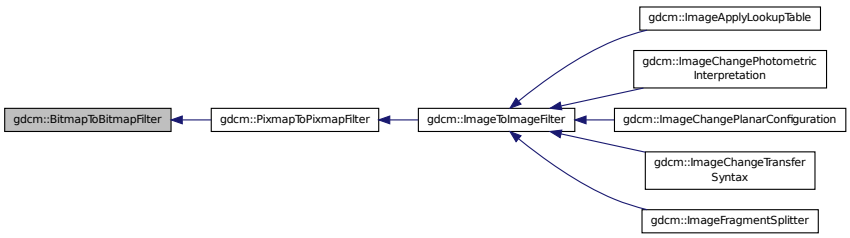
- gdcmBitmap.h

25.34 gdcm::BitmapToBitmapFilter Class Reference

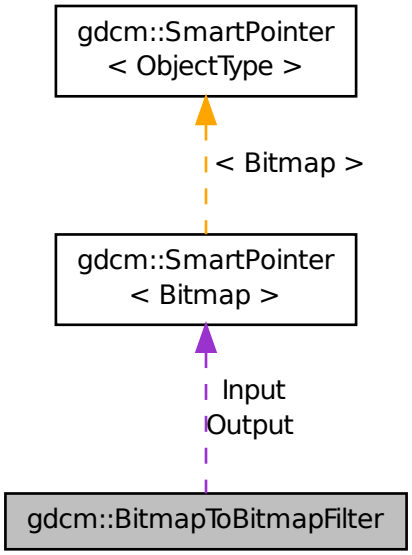
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.

`#include <gdcmBitmapToBitmapFilter.h>`

Inheritance diagram for gdcm::BitmapToBitmapFilter:



Collaboration diagram for gdcm::BitmapToBitmapFilter:



Public Member Functions

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()
- const Bitmap & GetOutput () const

  Get Output image.
- void SetInput (const Bitmap &image)

  Set input image.

Protected Attributes

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

25.34.1 Detailed Description

BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.

25.34.2 Constructor & Destructor Documentation

25.34.2.1 gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ( )

25.34.2.2 gdcm::BitmapToBitmapFilter::~BitmapToBitmapFilter ( ) [inline]

25.34.3 Member Function Documentation

25.34.3.1 const Bitmap& gdcm::BitmapToBitmapFilter::GetOutput ( ) const [inline]

Get Output image.

Reimplemented in gdcm::ImageToImageFilter, and gdcm::PixmapToPixmapFilter.

25.34.3.2 void gdcm::BitmapToBitmapFilter::SetInput (const Bitmap & *image*)

Set input image.

Examples:

CompressImage.cxx.

25.34.4 Member Data Documentation

25.34.4.1 SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input [protected]

25.34.4.2 SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output [protected]

The documentation for this class was generated from the following file:

- gdcmBitmapToBitmapFilter.h

25.35 gdcm::ByteBuffer Class Reference

ByteBuffer.

```
#include <gdcmByteBuffer.h>
```

Public Member Functions

- ByteBuffer ()
- char * Get (int len)
- const char * GetStart () const
- void ShiftEnd (int len)
- void UpdatePosition ()

25.35.1 Detailed Description

ByteBuffer.

Detailled description here

Note

looks like a std::streambuf or std::filebuf class with the get and peek pointer

25.35.2 Constructor & Destructor Documentation

25.35.2.1 gdcm::ByteBuffer::ByteBuffer ( ) [inline]

25.35.3 Member Function Documentation

25.35.3.1 char* gdcm::ByteBuffer::Get (int *len*) [inline]

25.35.3.2 const char* gdcm::ByteBuffer::GetStart ( ) const [inline]

25.35.3.3 void gdcm::ByteBuffer::ShiftEnd (int *len*) [inline]

25.35.3.4 void gdcm::ByteBuffer::UpdatePosition ( ) [inline]

The documentation for this class was generated from the following file:

- gdcmByteBuffer.h

25.36 gdcm::ByteSwap< T > Class Template Reference

ByteSwap.

```
#include <gdcmByteSwap.h>
```

Static Public Member Functions

- static void Swap (T &p)
- static void SwapFromSwapCodeIntoSystem (T &p, SwapCode const &sc)
- static void SwapRange (T *p, unsigned int num)
- static void SwapRangeFromSwapCodeIntoSystem (T *p, SwapCode const &sc, std::streamoff num)
- static bool SystemIsBigEndian ()
- static bool SystemIsLittleEndian ()

25.36.1 Detailed Description

template<class T>class gdcm::ByteSwap< T >

ByteSwap.

Perform machine dependent byte swaping (Little Endian, Big Endian, Bad Little Endian, Bad Big Endian). TODO: bswap_32 / bswap_64 ...

Examples:

TestByteSwap.cxx.

25.36.2 Member Function Documentation

25.36.2.1 template<class T > static void gdcm::ByteSwap< T >::Swap (T & *p*) [static]

25.36.2.2 template<class T> static void gdcm::ByteSwap<T>::SwapFromSwapCodeIntoSystem (T & *p*, SwapCode const & *sc*) [static]

Examples:

TestByteSwap.cxx.

25.36.2.3 template<class T> static void gdcm::ByteSwap<T>::SwapRange (T * *p*, unsigned int *num*) [static]

25.36.2.4 template<class T> static void gdcm::ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem (T * *p*, SwapCode const & *sc*, std::streamoff *num*) [static]

Examples:

TestByteSwap.cxx.

25.36.2.5 template<class T> static bool gdcm::ByteSwap<T>::SystemIsBigEndian ( ) [static]

Query the machine Endian-ness.

25.36.2.6 template<class T> static bool gdcm::ByteSwap<T>::SystemIsLittleEndian ( ) [static]

The documentation for this class was generated from the following file:

- gdcmByteSwap.h

25.37 gdcm::ByteSwapFilter Class Reference

ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??

```
#include <gdcmByteSwapFilter.h>
```

Public Member Functions

- ByteSwapFilter (DataSet &ds)
- ~ByteSwapFilter ()
- bool ByteSwap ()
- void SetByteSwapTag (bool b)

25.37.1 Detailed Description

ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??

25.37.2 Constructor & Destructor Documentation

25.37.2.1 gdcm::ByteSwapFilter::ByteSwapFilter (DataSet & *ds*) [inline]

25.37.2.2 gdcm::ByteSwapFilter::~ByteSwapFilter ( )

25.37.3 Member Function Documentation

25.37.3.1 bool gdcm::ByteSwapFilter::ByteSwap ( )

25.37.3.2 void gdcm::ByteSwapFilter::SetByteSwapTag (bool *b*) [inline]

The documentation for this class was generated from the following file:

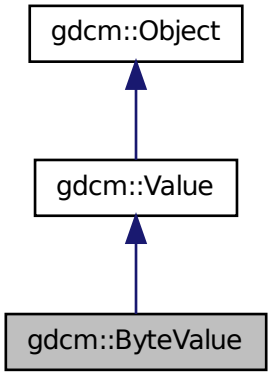
- gdcmByteSwapFilter.h

25.38 gdcm::ByteValue Class Reference

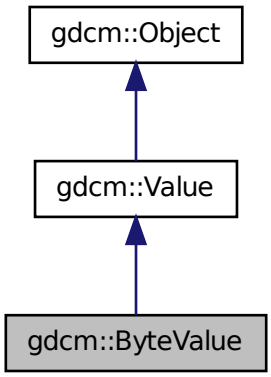
Class to represent binary value (array of bytes)

`#include <gdcmByteValue.h>`

Inheritance diagram for gdcm::ByteValue:



Collaboration diagram for gdcm::ByteValue:



Public Member Functions

- ByteValue (const char ∗array=0, VL const &vl=0)
- ByteValue (std::vector< char > &v)
- ∼ByteValue ()
- void Clear ()
- void Fill (char c)
- bool GetBuffer (char ∗buffer, unsigned long length) const
- VL GetLength () const
- const char ∗ GetPointer () const
- bool IsEmpty () const
- bool IsPrintable (VL length) const

  Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) I dont think this function is working since it does not handle UNICODE or character set...
- operator const std::vector< char > & () const
- ByteValue & operator= (const ByteValue &val)
- bool operator== (const ByteValue &val) const
- bool operator== (const Value &val) const
- void PrintASCII (std::ostream &os, VL maxlength) const
- void PrintGroupLength (std::ostream &os)
- void PrintHex (std::ostream &os, VL maxlength) const
- template<typename TSwap , typename TType >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- void SetLength (VL vl)
- template<typename TSwap , typename TType >
  std::ostream const & Write (std::ostream &os) const
- template<typename TSwap >
  std::ostream const & Write (std::ostream &os) const
- bool WriteBuffer (std::ostream &os) const

Protected Member Functions

- void Print (std::ostream &os) const

25.38.1 Detailed Description

Class to represent binary value (array of bytes)

Note

Examples:

DumpADAC.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, and rle2img.cxx.

25.38.2 Constructor & Destructor Documentation

25.38.2.1 gdcm::ByteValue::ByteValue (const char * *array* = 0, VL const & *vl* = 0) [inline]

References gdcmDebugMacro.

25.38.2.2 gdcm::ByteValue::ByteValue (std::vector< char > & *v*) [inline]

Warning

casting to uint32_t

25.38.2.3 gdcm::ByteValue::~ByteValue ( ) [inline]

25.38.3 Member Function Documentation

25.38.3.1 void gdcm::ByteValue::Clear ( ) [inline], [virtual]

Implements gdcm::Value.

25.38.3.2 void gdcm::ByteValue::Fill (char *c*) [inline]

Examples:

DuplicatePCDE.cxx.

25.38.3.3 bool gdcm::ByteValue::GetBuffer (char * *buffer*, unsigned long *length*) const

Examples:

FixJAIBugJPEGLS.cxx.

25.38.3.4 VL gdcm::ByteValue::GetLength ( ) const [inline], [virtual]

Implements gdcm::Value.

Examples:

DumpADAC.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), gdcm::Element< VR::OB, VM::VM1_n >::Set(), gdcm::Element< TVR, VM::VM1_n >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< VR::OB, VM::VM1_n >::SetNoSwap(), gdcm::Element< TVR, VM::VM1_n >::SetNoSwap(), and gdcm::Fragment::Write().

25.38.3.5 const char* gdcm::ByteValue::GetPointer ( ) const [inline]

Examples:

DumpADAC.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), gdcm::Element< VR::OB, VM::VM1_n >::Set(), gdcm::Element< TVR, VM::VM1_n >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< VR::OB, VM::VM1_n >::SetNoSwap(), and gdcm::Element< TVR, VM::VM1_n >::SetNoSwap().

25.38.3.6 bool gdcm::ByteValue::IsEmpty ( ) const [inline]

25.38.3.7 bool gdcm::ByteValue::IsPrintable (VL *length*) const [inline]

Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) I dont think this function is working since it does not handle UNICODE or character set...

25.38.3.8 gdcm::ByteValue::operator const std::vector< char > & ( ) const [inline]

25.38.3.9 ByteValue& gdcm::ByteValue::operator= (const ByteValue & *val*) [inline]

25.38.3.10 bool gdcm::ByteValue::operator== (const ByteValue & *val*) const [inline]

25.38.3.11 bool gdcm::ByteValue::operator== (const Value & *val*) const [inline], [virtual]

Implements gdcm::Value.

25.38.3.12 void gdcm::ByteValue::Print (std::ostream & *os*) const [inline], [protected], [virtual]

Reimplemented from gdcm::Object.

25.38.3.13 void gdcm::ByteValue::PrintASCII (std::ostream & *os*, VL *maxlength*) const

25.38.3.14 void gdcm::ByteValue::PrintGroupLength (std::ostream & *os*) [inline]

25.38.3.15 void gdcm::ByteValue::PrintHex (std::ostream & *os*, VL *maxlength*) const

25.38.3.16 template<typename TSwap , typename TType > std::istream& gdcm::ByteValue::Read (std::istream & *is*) [inline]

25.38.3.17 template<typename TSwap > std::istream& gdcm::ByteValue::Read (std::istream & *is*) [inline]

25.38.3.18 void gdcm::ByteValue::SetLength (VL *vl*) [inline], [virtual]

Implements gdcm::Value.

References gdcmDebugMacro, gdcm::VL::IsOdd(), and gdcm::VL::IsUndefined().

25.38.3.19 template<typename TSwap , typename TType > std::ostream const& gdcm::ByteValue::Write (std::ostream & *os*) const [inline]

Referenced by gdcm::Fragment::Write().

25.38.3.20 template<typename TSwap > std::ostream const& gdcm::ByteValue::Write (std::ostream & *os*) const [inline]

25.38.3.21 bool gdcm::ByteValue::WriteBuffer (std::ostream & *os*) const [inline]

The documentation for this class was generated from the following file:

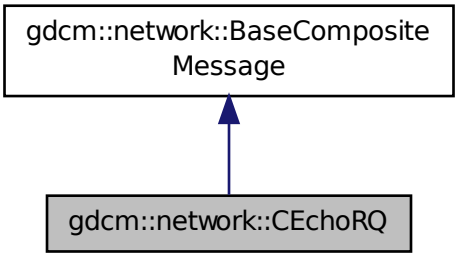
- gdcmByteValue.h

25.39 gdcm::network::CEchoRQ Class Reference

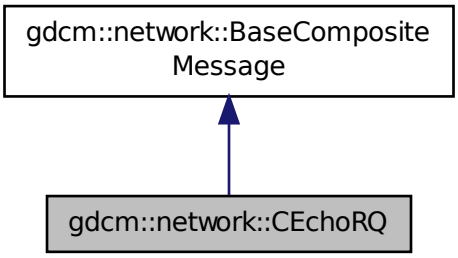
CEchoRQ this file defines the messages for the cecho action.

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for gdcm::network::CEchoRQ:



Collaboration diagram for gdcm::network::CEchoRQ:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)

Public Attributes

- UIComp AffectedSOPClassUID
- uint16_t MessageID

25.39.1 Detailed Description

CEchoRQ this file defines the messages for the cecho action.

25.39.2 Member Function Documentation

25.39.2.1 std::vector<PresentationDataValue> gdcm::network::CEchoRQ::ConstructPDV (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [virtual]

Implements gdcm::network::BaseCompositeMessage.

25.39.3 Member Data Documentation

25.39.3.1 UIComp gdcm::network::CEchoRQ::AffectedSOPClassUID

25.39.3.2 uint16_t gdcm::network::CEchoRQ::MessageID

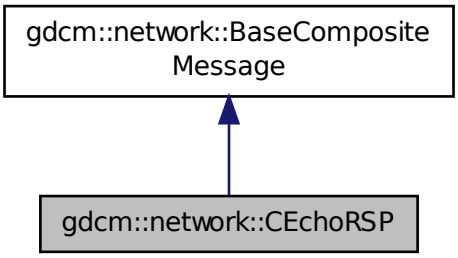
The documentation for this class was generated from the following files:

- gdcmCEchoMessages.h
- gdcmDIMSE.h

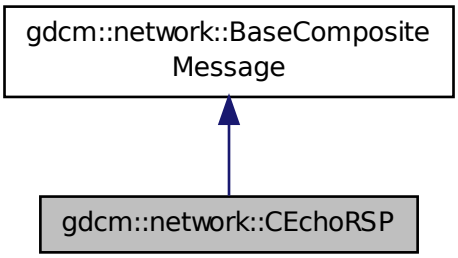
25.40 gdcm::network::CEchoRSP Class Reference

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for gdcm::network::CEchoRSP:



Collaboration diagram for gdcm::network::CEchoRSP:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const DataSet *inDataSet)

25.40.1 Member Function Documentation

25.40.1.1 std::vector<PresentationDataValue> gdcm::network::CEchoRSP::ConstructPDV (const DataSet * *inDataSet*)

The documentation for this class was generated from the following file:

- gdcmCEchoMessages.h

25.41 gdcm::network::CFind Class Reference

```
#include <gdcmDIMSE.h>
```

25.41.1 Detailed Description

PS 3.4 - 2009 Table B.2-1 C-STORE STATUS

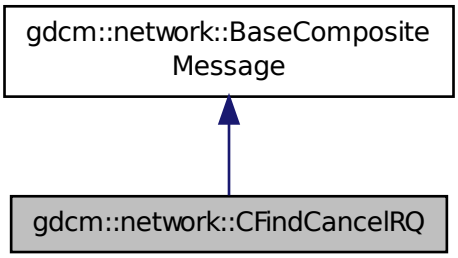
The documentation for this class was generated from the following file:

- gdcmDIMSE.h

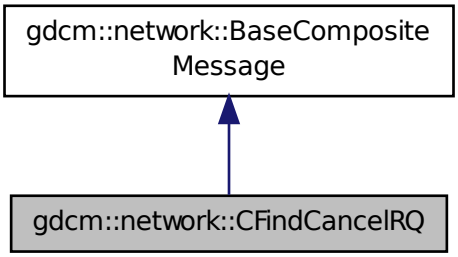
25.42 gdcm::network::CFindCancelRQ Class Reference

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for gdcm::network::CFindCancelRQ:



Collaboration diagram for gdcm::network::CFindCancelRQ:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const DataSet *inDataSet)

25.42.1 Member Function Documentation

25.42.1.1 std::vector<PresentationDataValue> gdcm::network::CFindCancelRQ::ConstructPDV (const DataSet * *inDataSet*)

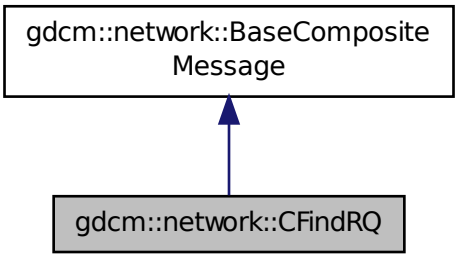
The documentation for this class was generated from the following file:

- gdcmCFindMessages.h

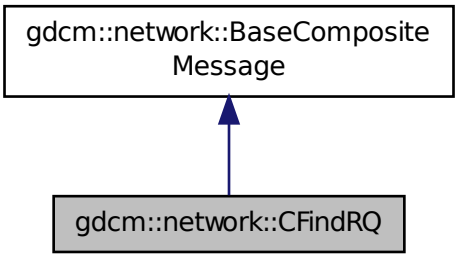
25.43 gdcm::network::CFindRQ Class Reference

`#include <gdcmCFindMessages.h>`

Inheritance diagram for gdcm::network::CFindRQ:



Collaboration diagram for gdcm::network::CFindRQ:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)

25.43.1 Member Function Documentation

25.43.1.1 std::vector<PresentationDataValue> gdcm::network::CFindRQ::ConstructPDV (const ULConnection & *inConnection*, const BaseRootQuery * *inRootQuery*) [virtual]

Implements gdcm::network::BaseCompositeMessage.

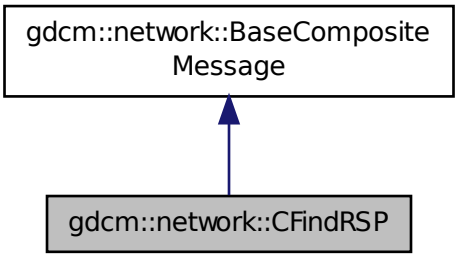
The documentation for this class was generated from the following file:

- gdcmCFindMessages.h

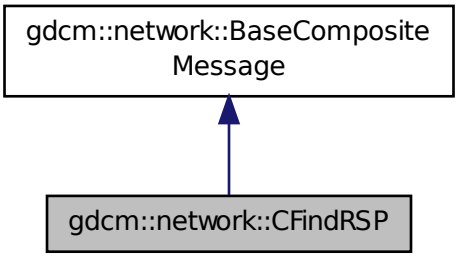
25.44 gdcm::network::CFindRSP Class Reference

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for gdcm::network::CFindRSP:



Collaboration diagram for gdcm::network::CFindRSP:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const DataSet *inDataSet)

25.44.1 Member Function Documentation

25.44.1.1 std::vector<PresentationDataValue> gdcm::network::CFindRSP::ConstructPDV (const DataSet * *inDataSet*)

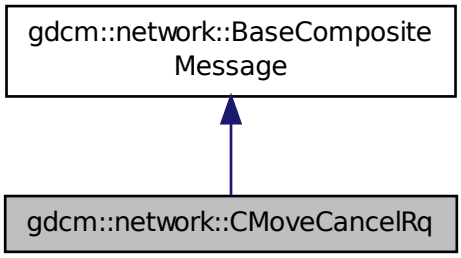
The documentation for this class was generated from the following file:

- gdcmCFindMessages.h

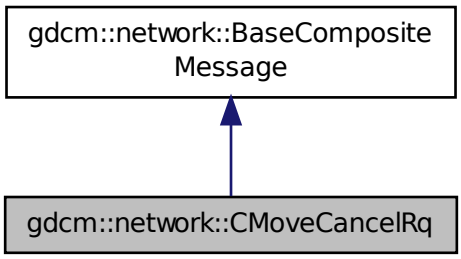
25.45 gdcm::network::CMoveCancelRq Class Reference

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveCancelRq:



Collaboration diagram for gdcm::network::CMoveCancelRq:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const DataSet *inDataSet)

25.45.1 Member Function Documentation

25.45.1.1 std::vector<PresentationDataValue> gdcm::network::CMoveCancelRq::ConstructPDV (const DataSet * *inDataSet*)

The documentation for this class was generated from the following file:

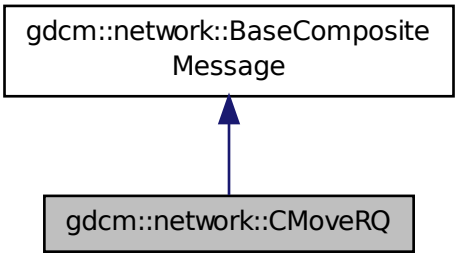
- gdcmCMoveMessages.h

25.46 gdcm::network::CMoveRQ Class Reference

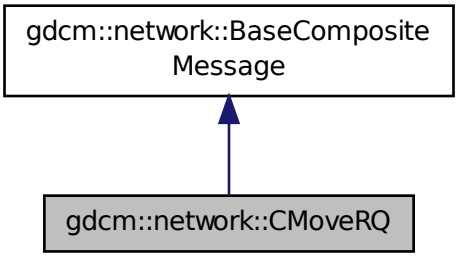
CMoveRQ this file defines the messages for the cmove action.

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveRQ:



Collaboration diagram for gdcm::network::CMoveRQ:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)

25.46.1 Detailed Description

CMoveRQ this file defines the messages for the cmove action.

25.46.2 Member Function Documentation

25.46.2.1 std::vector<PresentationDataValue> gdcm::network::CMoveRQ::ConstructPDV (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [virtual]

Implements gdcm::network::BaseCompositeMessage.

The documentation for this class was generated from the following file:

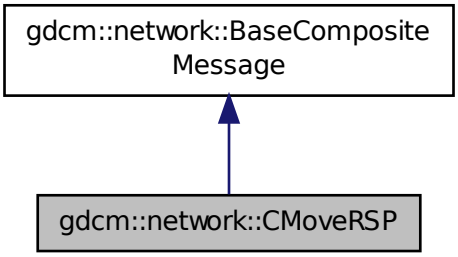
- gdcmCMoveMessages.h

25.47 gdcm::network::CMoveRSP Class Reference

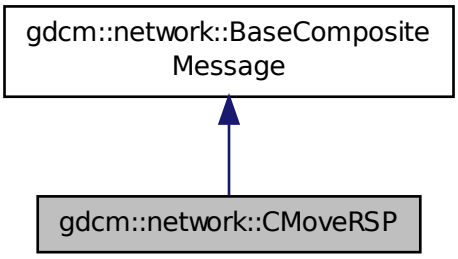
CMoveRSP this file defines the messages for the cmove action.

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveRSP:



Collaboration diagram for gdcm::network::CMoveRSP:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const DataSet *inDataSet)

25.47.1 Detailed Description

CMoveRSP this file defines the messages for the cmove action.

25.47.2 Member Function Documentation

25.47.2.1 std::vector<PresentationDataValue> gdcm::network::CMoveRSP::ConstructPDV (const DataSet * *inDataSet*)

The documentation for this class was generated from the following file:

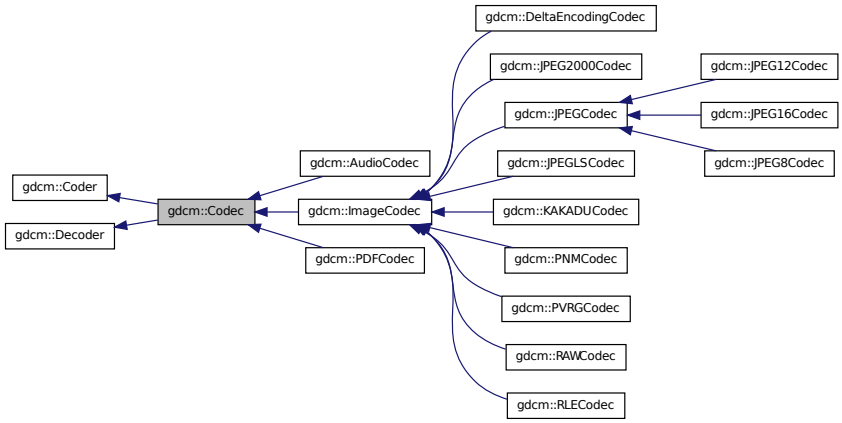
- gdcmCMoveMessages.h

25.48 gdcm::Codec Class Reference

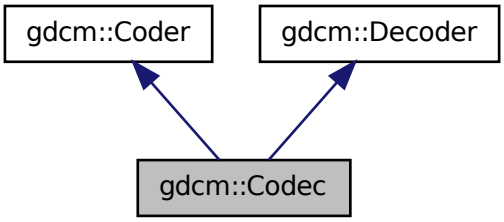
Codec class.

`#include <gdcmCodec.h>`

Inheritance diagram for gdcm::Codec:



Collaboration diagram for gdcm::Codec:



Additional Inherited Members

25.48.1 Detailed Description

Codec class.

The documentation for this class was generated from the following file:

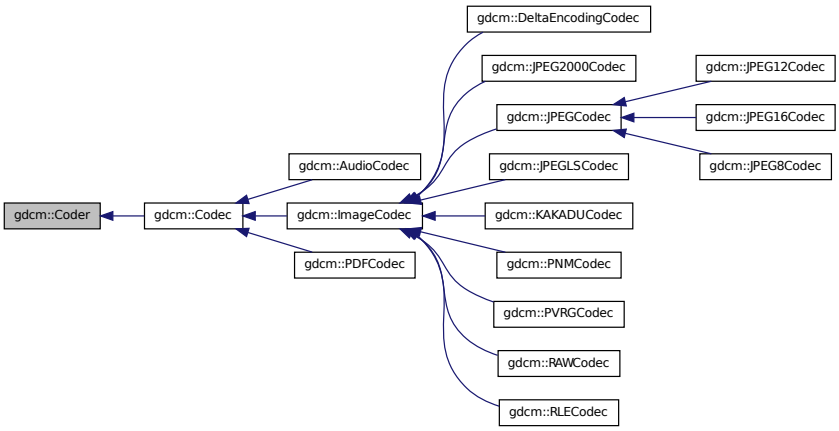
- gdcmCodec.h

25.49 gdcm::Coder Class Reference

Coder.

```
#include <gdcmCoder.h>
```

Inheritance diagram for gdcm::Coder:



Public Member Functions

- virtual ∼Coder ()
- virtual bool CanCode (TransferSyntax const &) const =0

  Return whether this coder support this transfer syntax (can code it)
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Protected Member Functions

- virtual bool InternalCode (const char ∗bv, unsigned long len, std::ostream &os)

25.49.1 Detailed Description

Coder.

25.49.2 Constructor & Destructor Documentation

25.49.2.1 virtual gdcm::Coder::∼Coder ( ) [inline],[virtual]

25.49.3 Member Function Documentation

25.49.3.1 virtual bool gdcm::Coder::CanCode (TransferSyntax const &) const [pure virtual]

Return whether this coder support this transfer syntax (can code it)

Implemented in gdcm::JPEGCodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::ImageCodec, gdcm::JPEGLSCodec, gdcm::PNMCodec, gdcm::KAKADUCodec, gdcm::RAWCodec, gdcm::AudioCodec, and gdcm::PDFCodec.

25.49.3.2 virtual bool gdcm::Coder::Code (DataElement const & *in_*, DataElement & *out_*) [inline],[virtual]

Code.

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

25.49.3.3 virtual bool gdcm::Coder::InternalCode (const char ∗ *bv*, unsigned long *len*, std::ostream & *os*) [inline], [protected],[virtual]

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

The documentation for this class was generated from the following file:

- gdcmCoder.h

25.50 gdcm::CodeString Class Reference

CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

```
#include <gdcmCodeString.h>
```

Public Types

- typedef InternalClass::const_iterator const_iterator
- typedef InternalClass::const_reference const_reference
- typedef InternalClass::const_reverse_iterator const_reverse_iterator
- typedef InternalClass::difference_type difference_type
- typedef InternalClass::iterator iterator
- typedef InternalClass::pointer pointer
- typedef InternalClass::reference reference
- typedef InternalClass::reverse_iterator reverse_iterator
- typedef InternalClass::size_type size_type
- typedef InternalClass::value_type value_type

Public Member Functions

- CodeString ()

  CodeString constructors.
- CodeString (const value_type ∗s)
- CodeString (const value_type ∗s, size_type n)
- CodeString (const InternalClass &s, size_type pos=0, size_type n=InternalClass::npos)
- std::string GetAsString () const

  Return the full code string as std::string.
- bool IsValid () const

  Check if CodeString obj is correct..
- size_type Size () const

  Return the size of the string.

Protected Member Functions

- std::string TrimInternal () const

Friends

- bool operator!= (const CodeString &ref, const CodeString &cs)
- std::ostream & operator<< (std::ostream &os, const CodeString &str)
- bool operator== (const CodeString &ref, const CodeString &cs)

25.50.1 Detailed Description

CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

Note

the cstor of CodeString will Trim the string on the fly so as to remove the extra leading and ending spaces. However it will not perform validation on the fly (CodeString obj can contains invalid char such as lower cases). This design was chosen to be a little tolerant to broken DICOM implementation, and thus allow user to compare lower case CS from there input file without the need to first rewrite them to get rid of invalid character (validation is a different operation from searching, querying).

Warning

when writing out DICOM file it is highly recommended to perform the IsValid() call, at least to check that the length of the string match the definition in the standard.

25.50.2 Member Typedef Documentation

25.50.2.1 typedef InternalClass::const_iterator gdcm::CodeString::const_iterator

25.50.2.2 typedef InternalClass::const_reference gdcm::CodeString::const_reference

25.50.2.3 typedef InternalClass::const_reverse_iterator gdcm::CodeString::const_reverse_iterator

25.50.2.4 typedef InternalClass::difference_type gdcm::CodeString::difference_type

25.50.2.5 typedef InternalClass::iterator gdcm::CodeString::iterator

25.50.2.6 typedef InternalClass::pointer gdcm::CodeString::pointer

25.50.2.7 typedef InternalClass::reference gdcm::CodeString::reference

25.50.2.8 typedef InternalClass::reverse_iterator gdcm::CodeString::reverse_iterator

25.50.2.9 typedef InternalClass::size_type gdcm::CodeString::size_type

25.50.2.10 typedef InternalClass::value_type gdcm::CodeString::value_type

25.50.3 Constructor & Destructor Documentation

25.50.3.1 gdcm::CodeString::CodeString ( ) [inline]

CodeString constructors.

25.50.3.2 gdcm::CodeString::CodeString (const value_type * *s*) [inline]

25.50.3.3 gdcm::CodeString::CodeString (const value_type * *s*, size_type *n*) [inline]

25.50.3.4 gdcm::CodeString::CodeString (const InternalClass & *s*, size_type *pos* = 0, size_type *n* = InternalClass::npos) [inline]

25.50.4 Member Function Documentation

25.50.4.1 std::string gdcm::CodeString::GetAsString ( ) const [inline]

Return the full code string as std::string.

25.50.4.2 bool gdcm::CodeString::IsValid ( ) const

Check if CodeString obj is correct..

25.50.4.3 size_type gdcm::CodeString::Size ( ) const [inline]

Return the size of the string.

25.50.4.4 std::string gdcm::CodeString::TrimInternal ( ) const [inline], [protected]

25.50.5 Friends And Related Function Documentation

25.50.5.1 bool operator!= (const CodeString & *ref*, const CodeString & *cs*) [friend]

25.50.5.2 std::ostream& operator<< (std::ostream & *os*, const CodeString & *str*) [friend]

25.50.5.3 bool operator== (const CodeString & *ref*, const CodeString & *cs*) [friend]

The documentation for this class was generated from the following file:

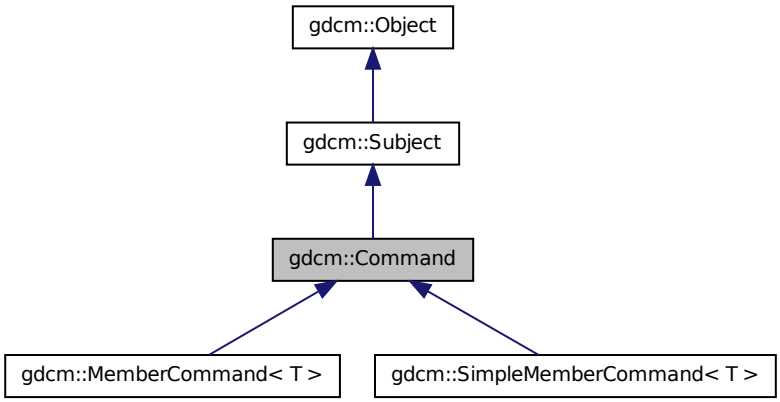
- gdcmCodeString.h

25.51 gdcm::Command Class Reference

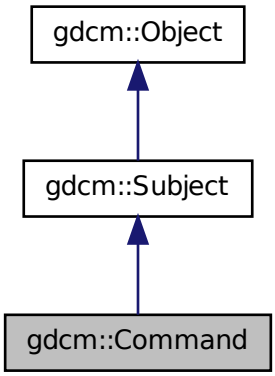
Command superclass for callback/observer methods.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::Command:



Collaboration diagram for gdcm::Command:



Public Member Functions

- virtual void Execute (Subject ∗caller, const Event &event)=0

  Abstract method that defines the action to be taken by the command.
- virtual void Execute (const Subject ∗caller, const Event &event)=0

Protected Member Functions

- Command ()

- ~Command ()

25.51.1 Detailed Description

Command superclass for callback/observer methods.

See also

Subject

25.51.2 Constructor & Destructor Documentation

25.51.2.1 gdcm::Command::Command ( ) [protected]

25.51.2.2 gdcm::Command::~Command ( ) [protected]

25.51.3 Member Function Documentation

25.51.3.1 virtual void gdcm::Command::Execute (Subject * *caller,* const Event & *event*) [pure virtual]

Abstract method that defines the action to be taken by the command.

Implemented in gdcm::SimpleMemberCommand< T >, and gdcm::MemberCommand< T >.

25.51.3.2 virtual void gdcm::Command::Execute (const Subject * *caller,* const Event & *event*) [pure virtual]

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const Object

Implemented in gdcm::SimpleMemberCommand< T >, and gdcm::MemberCommand< T >.

The documentation for this class was generated from the following file:

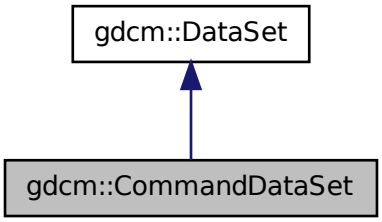
- gdcmCommand.h

25.52 gdcm::CommandDataSet Class Reference

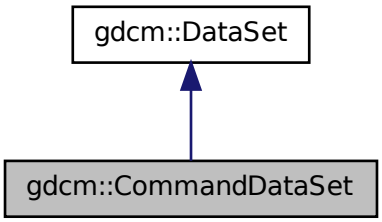
Class to represent a Command DataSet.

```
#include <gdcmCommandDataSet.h>
```

Inheritance diagram for gdcm::CommandDataSet:



Collaboration diagram for gdcm::CommandDataSet:



Public Member Functions

- CommandDataSet ()
- ~CommandDataSet ()
- void Insert (const DataElement &de)
- std::istream & Read (std::istream &is)

  Read.
- void Replace (const DataElement &de)

  Replace a dataelement with another one.
- std::ostream & Write (std::ostream &os) const

  Write.

Friends

- std::ostream & operator<< (std::ostream &_os, const CommandDataSet &_val)

Additional Inherited Members

25.52.1 Detailed Description

Class to represent a Command DataSet.

See also

DataSet

25.52.2 Constructor & Destructor Documentation

25.52.2.1 gdcm::CommandDataSet::CommandDataSet ( ) [inline]

25.52.2.2 gdcm::CommandDataSet::~CommandDataSet ( ) [inline]

25.52.3 Member Function Documentation

25.52.3.1 void gdcm::CommandDataSet::Insert (const DataElement & *de*) [inline]

Insert a DataElement in the DataSet.

Warning

: Tag need to be >= 0x8 to be considered valid data element

Reimplemented from gdcm::DataSet.

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

25.52.3.2 std::istream& gdcm::CommandDataSet::Read (std::istream & *is*)

Read.

Reimplemented from gdcm::DataSet.

25.52.3.3 void gdcm::CommandDataSet::Replace (const DataElement & *de*) [inline]

Replace a dataelement with another one.

Reimplemented from gdcm::DataSet.

References gdcm::DataElement::GetTag().

25.52.3.4 std::ostream& gdcm::CommandDataSet::Write (std::ostream & *os*) const

Write.

Reimplemented from gdcm::DataSet.

25.52.4 Friends And Related Function Documentation

25.52.4.1 std::ostream& operator<< (std::ostream & _os, const CommandDataSet & _val) [friend]

The documentation for this class was generated from the following file:

- gdcmCommandDataSet.h

25.53 gdcm::network::CompositeMessageFactory Class Reference

CompositeMessageFactory This class constructs PDataPDUs, but that have been specifically constructed for the composite DICOM services (C-Echo, C-Find, C-Get, C-Move, and C-Store). It will also handle parsing the incoming data to determine which of the CompositePDUs the incoming data is, and so therefore allowing the scu to determine what to do with incoming data (if acting as a storescp server, for instance).

```
#include <gdcmCompositeMessageFactory.h>
```

Static Public Member Functions

- static std::vector < PresentationDataValue > ConstructCEchoRQ (const ULConnection &inConnection)
- static std::vector < PresentationDataValue > ConstructCFindRQ (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector < PresentationDataValue > ConstructCMoveRQ (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector < PresentationDataValue > ConstructCStoreRQ (const ULConnection &inConnection, const File &file)
- static std::vector < PresentationDataValue > ConstructCStoreRSP (const DataSet *inDataSet, const BasePDU *inPC)

25.53.1 Detailed Description

CompositeMessageFactory This class constructs PDataPDUs, but that have been specifically constructed for the composite DICOM services (C-Echo, C-Find, C-Get, C-Move, and C-Store). It will also handle parsing the incoming data to determine which of the CompositePDUs the incoming data is, and so therefore allowing the scu to determine what to do with incoming data (if acting as a storescp server, for instance).

25.53.2 Member Function Documentation

25.53.2.1 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCEchoRQ (const ULConnection & inConnection) [static]

25.53.2.2 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCFindRQ (const ULConnection & inConnection, const BaseRootQuery * inRootQuery) [static]

25.53.2.3 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCMoveRQ (const ULConnection & inConnection, const BaseRootQuery * inRootQuery) [static]

25.53.2.4 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCStoreRQ (const ULConnection & *inConnection,* const File & *file*) [static]

25.53.2.5 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCStoreRSP (const DataSet * *inDataSet,* const BasePDU * *inPC*) [static]

The documentation for this class was generated from the following file:

- gdcmCompositeMessageFactory.h

25.54 gdcm::CompositeNetworkFunctions Class Reference

Composite Network Functions These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

```
#include <gdcmCompositeNetworkFunctions.h>
```

Public Types

- typedef std::vector
  < KeyValuePairType > KeyValuePairArrayType
- typedef std::pair< Tag,
  std::string > KeyValuePairType

Static Public Member Functions

- static bool CEcho (const char *remote, uint16_t portno, const char *aetitle=NULL, const char *call=NULL)
- static bool CFind (const char *remote, uint16_t portno, const BaseRootQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle=NULL, const char *call=NULL)
- static bool CMove (const char *remote, uint16_t portno, const BaseRootQuery *query, uint16_t portscp, const char *aetitle=NULL, const char *call=NULL, const char *outputdir=NULL)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const DataSet &queryds, bool inMove=false)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValue-PairArrayType &keys, bool inMove=false)
- static bool CStore (const char *remote, uint16_t portno, const Directory::FilenamesType &filenames, const char *aetitle=NULL, const char *call=NULL)

25.54.1 Detailed Description

Composite Network Functions These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program.

The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

- C-ECHO SCU
- C-FIND SCU
- C-STORE SCU
- C-MOVE SCU (+internal C-STORE SCP)

25.54.2 Member Typedef Documentation

25.54.2.1 typedef std::vector< KeyValuePairType > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType

25.54.2.2 typedef std::pair<Tag, std::string> gdcm::CompositeNetworkFunctions::KeyValuePairType

25.54.3 Member Function Documentation

25.54.3.1 static bool gdcm::CompositeNetworkFunctions::CEcho (const char * *remote*, uint16_t *portno*, const char * *aetitle* = NULL, const char * *call* = NULL) [static]

The most basic network function. Use this function to ensure that the remote server is responding on the given IP and port number as expected.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

25.54.3.2 static bool gdcm::CompositeNetworkFunctions::CFind (const char * *remote*, uint16_t *portno*, const BaseRootQuery * *query*, std::vector< DataSet > & *retDataSets*, const char * *aetitle* = NULL, const char * *call* = NULL) [static]

This function will use the provided query to determine what files a remote server contains that match the query strings. The return is a vector of datasets that contain tags as reported by the server. If the dataset is empty, then it is possible that an error condition was encountered; in which case, the user should monitor the error and warning streams.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

25.54.3.3 static bool gdcm::CompositeNetworkFunctions::CMove (const char * *remote*, uint16_t *portno*, const BaseRootQuery * *query*, uint16_t *portscp*, const char * *aetitle* = NULL, const char * *call* = NULL, const char * *outputdir* = NULL) [static]

This function will use the provided query to get files from a remote server. NOTE that this functionality is essentially equivalent to C-GET in the DICOM standard; however, C-GET has been deprecated, so this function allows for the user to ask a remote server for files matching a query and return them to the local machine. Files will be written to the given output directory. If the operation succeeds, the function returns true. This function is a prime candidate for being overwritten by expert users; if the datasets should remain in memory, for instance, that behavior can be changed by creating a user-level version of this function.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0 when
outputdir	is not set default to current dir ('.')

Returns

true if it worked.

25.54.3.4 static BaseRootQuery* gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType *inRootType*, EQueryLevel *inQueryLevel*, const DataSet & *queryds*, bool *inMove* = false) [static]

This function will take a list of strings and tags and fill in a query that can be used for either CFind or CMove (depending on the input boolean

Parameters

inMove).	Note that the caller is responsible for deleting the constructed query. This function is used to build both a move and a find query (true for inMove if it's move, false if it's find)

25.54.3.5 static BaseRootQuery* gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType *inRootType*, EQueryLevel *inQueryLevel*, const KeyValuePairArrayType & *keys*, bool *inMove* = false) [static]

Deprecated

25.54.3.6 static bool gdcm::CompositeNetworkFunctions::CStore (const char * *remote*, uint16_t *portno*, const Directory::FilenamesType & *filenames*, const char * *aetitle* = NULL, const char * *call* = NULL) [static]

This function will place the provided files into the remote server. The function returns true if it worked for all files.

Warning

the server side can refuse an association on a given file

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0

Returns

true if it worked for all files

The documentation for this class was generated from the following file:

- gdcmCompositeNetworkFunctions.h

25.55 gdcm::ConstCharWrapper Class Reference

Do not use me.

`#include <gdcmConstCharWrapper.h>`

Public Member Functions

- ConstCharWrapper (const char ∗i=0)
- operator const char ∗ () const

25.55.1 Detailed Description

Do not use me.

25.55.2 Constructor & Destructor Documentation

25.55.2.1 gdcm::ConstCharWrapper::ConstCharWrapper (const char ∗ *i* = 0) [inline]

25.55.3 Member Function Documentation

25.55.3.1 gdcm::ConstCharWrapper::operator const char ∗ ( ) const [inline]

The documentation for this class was generated from the following file:

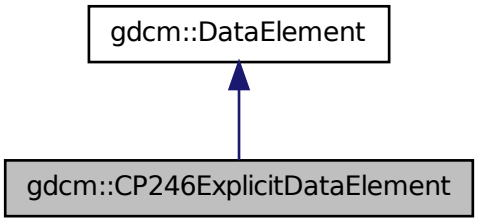
- gdcmConstCharWrapper.h

25.56 gdcm::CP246ExplicitDataElement Class Reference

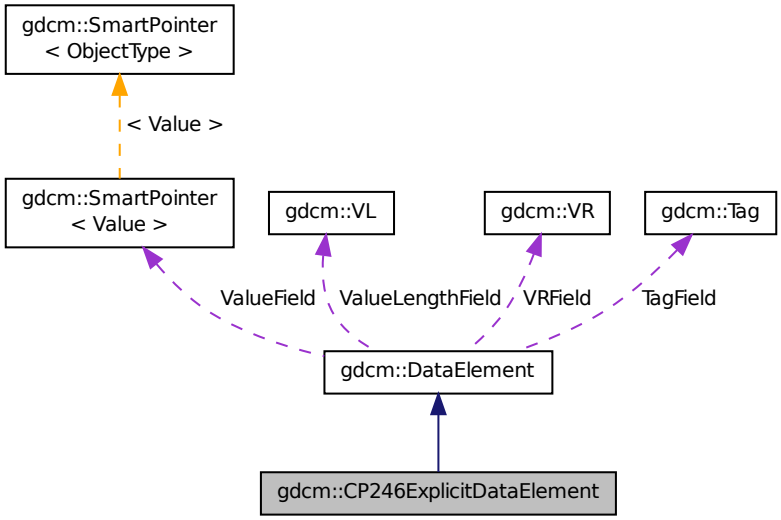
Class to read/write a DataElement as CP246Explicit Data Element.

`#include <gdcmCP246ExplicitDataElement.h>`

Inheritance diagram for gdcm::CP246ExplicitDataElement:



Collaboration diagram for gdcm::CP246ExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)

Additional Inherited Members

25.56.1 Detailed Description

Class to read/write a DataElement as CP246Explicit Data Element.

Note

Some system are producing SQ, declare them as UN, but encode the SQ as 'Explicit' instead of Implicit

25.56.2 Member Function Documentation

25.56.2.1 VL gdcm::CP246ExplicitDataElement::GetLength ( ) const

Reimplemented from gdcm::DataElement.

25.56.2.2 template<typename TSwap > std::istream& gdcm::CP246ExplicitDataElement::Read (std::istream & *is*)

Reimplemented from gdcm::DataElement.

25.56.2.3 template<typename TSwap > std::istream& gdcm::CP246ExplicitDataElement::ReadPreValue (std::istream & *is*)

25.56.2.4 template<typename TSwap > std::istream& gdcm::CP246ExplicitDataElement::ReadValue (std::istream & *is*)

25.56.2.5 template<typename TSwap > std::istream& gdcm::CP246ExplicitDataElement::ReadWithLength (std::istream & *is,* VL & *length*)

Reimplemented from gdcm::DataElement.

The documentation for this class was generated from the following file:

- gdcmCP246ExplicitDataElement.h

25.57 gdcm::CryptographicMessageSyntax Class Reference

Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

```
#include <gdcmCryptographicMessageSyntax.h>
```

Public Types

- enum CipherTypes {
  DES_CIPHER,
  DES3_CIPHER,
  AES128_CIPHER,
  AES192_CIPHER,
  AES256_CIPHER }

Public Member Functions

- CryptographicMessageSyntax ()
- ~CryptographicMessageSyntax ()
- bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const

  decrypt content from a PKCS#7 envelopedData structure
- bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const

  create a PKCS#7 envelopedData structure
- CipherTypes GetCipherType () const
- bool ParseCertificateFile (const char *filename)
- bool ParseKeyFile (const char *filename)
- void SetCipherType (CipherTypes type)

25.57.1 Detailed Description

Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

See online documentation `http://www.openssl.org/docs/crypto/PKCS7_encrypt.html`

25.57.2 Member Enumeration Documentation

25.57.2.1 enum gdcm::CryptographicMessageSyntax::CipherTypes

Enumerator:

DES_CIPHER

DES3_CIPHER

AES128_CIPHER

AES192_CIPHER

AES256_CIPHER

25.57.3 Constructor & Destructor Documentation

25.57.3.1 gdcm::CryptographicMessageSyntax::CryptographicMessageSyntax ( )

25.57.3.2 gdcm::CryptographicMessageSyntax::~CryptographicMessageSyntax ( )

25.57.4 Member Function Documentation

25.57.4.1 bool gdcm::CryptographicMessageSyntax::Decrypt (char * *output,* size_t & *outlen,* const char * *array,* size_t *len*) const

decrypt content from a PKCS#7 envelopedData structure

25.57.4.2 bool gdcm::CryptographicMessageSyntax::Encrypt (char * *output,* size_t & *outlen,* const char * *array,* size_t *len*) const

create a PKCS#7 envelopedData structure

25.57.4.3 CipherTypes gdcm::CryptographicMessageSyntax::GetCipherType ( ) const

25.57.4.4 bool gdcm::CryptographicMessageSyntax::ParseCertificateFile (const char * *filename*)

25.57.4.5 bool gdcm::CryptographicMessageSyntax::ParseKeyFile (const char * *filename*)

25.57.4.6 void gdcm::CryptographicMessageSyntax::SetCipherType (CipherTypes *type*)

Set Cipher Type. Default is: AES256_CIPHER

The documentation for this class was generated from the following file:

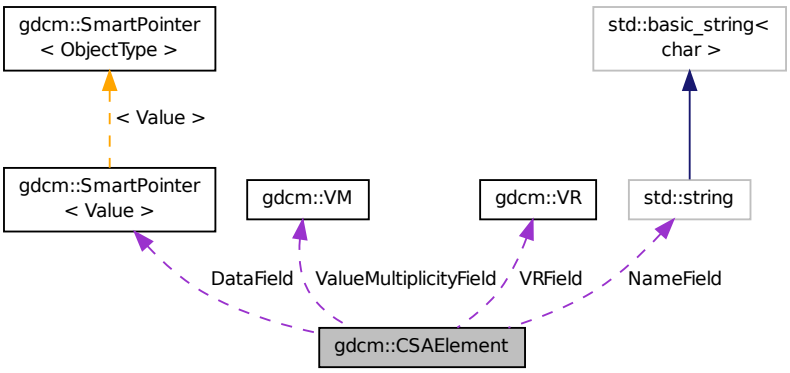
- gdcmCryptographicMessageSyntax.h

25.58 gdcm::CSAElement Class Reference

Class to represent a CSA Element.

`#include <gdcmCSAElement.h>`

Collaboration diagram for gdcm::CSAElement:



Public Member Functions

- CSAElement (unsigned int kf=0)
- CSAElement (const CSAElement &_val)
- const ByteValue * GetByteValue () const
- unsigned int GetKey () const

  Set/Get Key.
- const char * GetName () const

  Set/Get Name.
- unsigned int GetNoOfItems () const

  Set/Get NoOfItems.
- unsigned int GetSyngoDT () const

Set/Get SyngoDT.

- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- Value & GetValue ()
- const VM & GetVM () const

  Set/Get VM.
- VR const & GetVR () const

  Set/Get VR.
- bool IsEmpty () const

  Check if CSA Element is empty.
- bool operator< (const CSAElement &de) const
- CSAElement & operator= (const CSAElement &de)
- bool operator== (const CSAElement &de) const
- void SetByteValue (const char *array, VL length)

  Set.
- void SetKey (unsigned int key)
- void SetName (const char *name)
- void SetNoOfItems (unsigned int items)
- void SetSyngoDT (unsigned int syngodt)
- void SetValue (Value const &vl)
- void SetVM (const VM &vm)
- void SetVR (VR const &vr)

Protected Types

- typedef SmartPointer< Value > DataPtr

Protected Attributes

- DataPtr DataField
- unsigned int KeyField
- std::string NameField
- unsigned int NoOfItemsField
- unsigned int SyngoDTField
- VM ValueMultiplicityField
- VR VRField

Friends

- std::ostream & operator<< (std::ostream &os, const CSAElement &val)

25.58.1 Detailed Description

Class to represent a CSA Element.

See also

CSAHeader

Examples:

csa2img.cxx, and MrProtocol.cxx.

25.58.2 Member Typedef Documentation

25.58.2.1 typedef SmartPointer<Value> gdcm::CSAElement::DataPtr [protected]

25.58.3 Constructor & Destructor Documentation

25.58.3.1 gdcm::CSAElement::CSAElement (unsigned int *kf* = 0) [inline]

25.58.3.2 gdcm::CSAElement::CSAElement (const CSAElement & *_val*) [inline]

25.58.4 Member Function Documentation

25.58.4.1 const ByteValue* gdcm::CSAElement::GetByteValue ( ) const [inline]

Return the Value of CSAElement as a ByteValue (if possible)

Warning

: You need to check for NULL return value

Examples:

MrProtocol.cxx.

25.58.4.2 unsigned int gdcm::CSAElement::GetKey ( ) const [inline]

Set/Get Key.

Referenced by operator<().

25.58.4.3 const char* gdcm::CSAElement::GetName ( ) const [inline]

Set/Get Name.

25.58.4.4 unsigned int gdcm::CSAElement::GetNoOfItems ( ) const [inline]

Set/Get NoOfItems.

25.58.4.5 unsigned int gdcm::CSAElement::GetSyngoDT ( ) const [inline]

Set/Get SyngoDT.

25.58.4.6 Value const& gdcm::CSAElement::GetValue ( ) const [inline]

Set/Get Value (bytes array, SQ of items, SQ of fragments):

Examples:

csa2img.cxx.

25.58.4.7 Value& gdcm::CSAElement::GetValue ( ) [inline]

25.58.4.8 const VM& gdcm::CSAElement::GetVM ( ) const [inline]

Set/Get VM.

25.58.4.9 VR const& gdcm::CSAElement::GetVR ( ) const [inline]

Set/Get VR.

25.58.4.10 bool gdcm::CSAElement::IsEmpty ( ) const [inline]

Check if CSA Element is empty.

Examples:

csa2img.cxx.

25.58.4.11 bool gdcm::CSAElement::operator< (const CSAElement & *de*) const [inline]

References GetKey().

25.58.4.12 CSAElement& gdcm::CSAElement::operator= (const CSAElement & *de*) [inline]

References DataField, KeyField, NameField, NoOfItemsField, SyngoDTField, ValueMultiplicityField, and VRField.

25.58.4.13 bool gdcm::CSAElement::operator== (const CSAElement & *de*) const [inline]

References KeyField, NameField, SyngoDTField, ValueMultiplicityField, and VRField.

25.58.4.14 void gdcm::CSAElement::SetByteValue (const char * *array*, VL *length*) [inline]

Set.

25.58.4.15 void gdcm::CSAElement::SetKey (unsigned int *key*) [inline]

25.58.4.16 void gdcm::CSAElement::SetName (const char * *name*) [inline]

25.58.4.17 void gdcm::CSAElement::SetNoOfItems (unsigned int *items*) [inline]

25.58.4.18 void gdcm::CSAElement::SetSyngoDT (unsigned int *syngodt*) [inline]

25.58.4.19 void gdcm::CSAElement::SetValue (Value const & *vl*) [inline]

25.58.4.20 void gdcm::CSAElement::SetVM (const VM & *vm*) [inline]

25.58.4.21 void gdcm::CSAElement::SetVR (VR const & *vr*) [inline]

25.58.5 Friends And Related Function Documentation

25.58.5.1 std::ostream& operator<< (std::ostream & *os,* const CSAElement & *val*) [friend]

25.58.6 Member Data Documentation

25.58.6.1 DataPtr gdcm::CSAElement::DataField [protected]

Referenced by gdcm::operator<<(), and operator=().

25.58.6.2 unsigned int gdcm::CSAElement::KeyField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

25.58.6.3 std::string gdcm::CSAElement::NameField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

25.58.6.4 unsigned int gdcm::CSAElement::NoOfItemsField [protected]

Referenced by gdcm::operator<<(), and operator=().

25.58.6.5 unsigned int gdcm::CSAElement::SyngoDTField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

25.58.6.6 VM gdcm::CSAElement::ValueMultiplicityField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

25.58.6.7 VR gdcm::CSAElement::VRField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

The documentation for this class was generated from the following file:

- gdcmCSAElement.h

25.59 gdcm::CSAHeader Class Reference

Class for CSAHeader.

```
#include <gdcmCSAHeader.h>
```

Public Types

- enum CSAHeaderType {
  UNKNOWN = 0,
  SV10,
  NOMAGIC,
  DATASET_FORMAT,
  INTERFILE,
  ZEROED_OUT }

  Divers format of CSAHeader as found 'in the wild'.

Public Member Functions

- CSAHeader ()
- ∼CSAHeader ()
- bool FindCSAElementByName (const char ∗name)
- const CSAElement & GetCSAElementByName (const char ∗name)
- const DataSet & GetDataSet () const

  Return the DataSet output (use only if Format == DATASET_FORMAT)
- CSAHeaderType GetFormat () const
- const char ∗ GetInterfile () const

  Return the string output (use only if Format == Interfile)
- bool LoadFromDataElement (DataElement const &de)

  Decode the CSAHeader from element 'de'.
- void Print (std::ostream &os) const

  Print the CSAHeader (use only if Format == SV10 or NOMAGIC)
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

Static Public Member Functions

- static const PrivateTag & GetCSADataInfo ()
- static const PrivateTag & GetCSAImageHeaderInfoTag ()
- static const PrivateTag & GetCSASeriesHeaderInfoTag ()

Protected Member Functions

- const CSAElement & GetCSAEEnd () const

Friends

- std::ostream & operator<< (std::ostream &_os, const CSAHeader &d)

25.59.1 Detailed Description

Class for CSAHeader.

SIEMENS store private information in tag (0x0029,0x10,"SIEMENS CSA HEADER") this class is meant for user wishing to access values stored within this private attribute. There are basically two main 'format' for this attribute : SV10/NOMAGIC and DATASET_FORMAT SV10 and NOMAGIC are from a user prospective identical, see CSAHeader.xml for possible name / value stored in this format. DATASET_FORMAT is in fact simply just another DICOM dataset (implicit) with -currently unknown- value. This can be only be printed for now.

Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.
the API of this class might change.

Todo MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

See also

PDBHeader

External references: 5.1.3.2.4.1 MEDCOM History Information and 5.1.4.3 CSA Non-Image Module in `http://tamsinfo.toshiba.com/docrequest/pdf/E.Soft_v2.0.pdf`

Examples:

csa2img.cxx, and MrProtocol.cxx.

25.59.2 Member Enumeration Documentation

25.59.2.1 enum gdcm::CSAHeader::CSAHeaderType

Divers format of CSAHeader as found 'in the wild'.

Enumerator:

UNKNOWN
SV10
NOMAGIC
DATASET_FORMAT
INTERFILE
ZEROED_OUT

25.59.3 Constructor & Destructor Documentation

25.59.3.1 gdcm::CSAHeader::CSAHeader ( ) [inline]

25.59.3.2 gdcm::CSAHeader::~CSAHeader ( ) [inline]

25.59.4 Member Function Documentation

25.59.4.1 bool gdcm::CSAHeader::FindCSAElementByName (const char * *name*)

Return true if the CSA element matching 'name' is found or not

Warning

Case Sensitive

Examples:

csa2img.cxx, and MrProtocol.cxx.

25.59.4.2 static const PrivateTag& gdcm::CSAHeader::GetCSADataInfo ( ) [static]

Return the private tag used by SIEMENS to store the CSA Data Info This is: PrivateTag(0x0029,0x0010,"SIEMENS CSA NON-IMAGE");

25.59.4.3 const CSAElement& gdcm::CSAHeader::GetCSAEEnd ( ) const [protected]

25.59.4.4 const CSAElement& gdcm::CSAHeader::GetCSAElementByName (const char * *name*)

Return the CSAElement corresponding to name 'name'

Warning

Case Sensitive

Examples:

csa2img.cxx, and MrProtocol.cxx.

25.59.4.5 static const PrivateTag& gdcm::CSAHeader::GetCSAImageHeaderInfoTag ( ) [static]

Return the private tag used by SIEMENS to store the CSA Image Header This is: PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER");

Examples:

csa2img.cxx, and PublicDict.cxx.

25.59.4.6 static const PrivateTag& gdcm::CSAHeader::GetCSASeriesHeaderInfoTag ( ) [static]

Return the private tag used by SIEMENS to store the CSA Series Header This is: PrivateTag(0x0029,0x0020,"SIEMENS CSA HEADER");

Examples:

MrProtocol.cxx.

25.59.4.7 const DataSet& gdcm::CSAHeader::GetDataSet ( ) const [inline]

Return the DataSet output (use only if Format == DATASET_FORMAT)

25.59.4.8 CSAHeaderType gdcm::CSAHeader::GetFormat ( ) const

return the format of the CSAHeader SV10 and NOMAGIC are equivalent.

25.59.4.9 const char* gdcm::CSAHeader::GetInterfile ( ) const [inline]

Return the string output (use only if Format == Interfile)

25.59.4.10 bool gdcm::CSAHeader::LoadFromDataElement (DataElement const & *de*)

Decode the CSAHeader from element 'de'.

Examples:

csa2img.cxx, and MrProtocol.cxx.

25.59.4.11 void gdcm::CSAHeader::Print (std::ostream & *os*) const

Print the CSAHeader (use only if Format == SV10 or NOMAGIC)

Examples:

csa2img.cxx.

Referenced by gdcm::operator<<().

25.59.4.12 template<typename TSwap > std::istream& gdcm::CSAHeader::Read (std::istream & *is*)

25.59.4.13 template<typename TSwap > const std::ostream& gdcm::CSAHeader::Write (std::ostream & *os*) const

25.59.5 Friends And Related Function Documentation

25.59.5.1 std::ostream& operator<< (std::ostream & *_os,* const CSAHeader & *d*) [friend]

The documentation for this class was generated from the following file:

- gdcmCSAHeader.h

25.60 gdcm::CSAHeaderDict Class Reference

Class to represent a map of CSAHeaderDictEntry.

```
#include <gdcmCSAHeaderDict.h>
```

Public Types

- typedef MapCSAHeaderDictEntry::const_iterator ConstIterator
- typedef MapCSAHeaderDictEntry::iterator Iterator
- typedef std::set < CSAHeaderDictEntry > MapCSAHeaderDictEntry

Public Member Functions

- CSAHeaderDict ()
- void AddCSAHeaderDictEntry (const CSAHeaderDictEntry &de)
- ConstIterator Begin () const
- ConstIterator End () const
- const CSAHeaderDictEntry & GetCSAHeaderDictEntry (const char *name) const
- bool IsEmpty () const

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const CSAHeaderDict &_val)

25.60.1 Detailed Description

Class to represent a map of CSAHeaderDictEntry.

Examples:

MrProtocol.cxx.

25.60.2 Member Typedef Documentation

25.60.2.1 typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator

25.60.2.2 typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator

25.60.2.3 typedef std::set<CSAHeaderDictEntry> gdcm::CSAHeaderDict::MapCSAHeaderDictEntry

25.60.3 Constructor & Destructor Documentation

25.60.3.1 gdcm::CSAHeaderDict::CSAHeaderDict ( ) [inline]

25.60.4 Member Function Documentation

25.60.4.1 void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (const CSAHeaderDictEntry & *de*) [inline]

25.60.4.2 ConstIterator gdcm::CSAHeaderDict::Begin ( ) const [inline]

25.60.4.3 ConstIterator gdcm::CSAHeaderDict::End ( ) const [inline]

25.60.4.4 const CSAHeaderDictEntry& gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (const char * *name*) const [inline]

Examples:

MrProtocol.cxx.

25.60.4.5 bool gdcm::CSAHeaderDict::IsEmpty ( ) const [inline]

25.60.4.6 void gdcm::CSAHeaderDict::LoadDefault ( ) [protected]

25.60.5 Friends And Related Function Documentation

25.60.5.1 friend class Dicts [friend]

25.60.5.2 std::ostream& operator<< (std::ostream & *_os,* const CSAHeaderDict & *_val*) [friend]

The documentation for this class was generated from the following file:

- gdcmCSAHeaderDict.h

25.61 gdcm::CSAHeaderDictEntry Class Reference

Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information.

```
#include <gdcmCSAHeaderDictEntry.h>
```

Public Member Functions

- CSAHeaderDictEntry (const char *name="", VR const &vr=VR::INVALID, VM const &vm=VM::VM0, const char *desc="")
- const char * GetDescription () const

  Set/Get Description.
- const char * GetName () const

  Set/Get Name.
- const VM & GetVM () const

  Set/Get VM.
- const VR & GetVR () const

  Set/Get VR.
- bool operator< (const CSAHeaderDictEntry &entry) const
- void SetDescription (const char *desc)
- void SetName (const char *name)

- void SetVM (VM const &vm)
- void SetVR (const VR &vr)

Friends

- std::ostream & operator<< (std::ostream &_os, const CSAHeaderDictEntry &_val)

25.61.1 Detailed Description

Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information.

Note

bla TODO FIXME: Need a PublicCSAHeaderDictEntry...indeed CSAHeaderDictEntry has a notion of retired which does not exist in PrivateCSAHeaderDictEntry...

See also

gdcm::Dict

Examples:

MrProtocol.cxx.

25.61.2 Constructor & Destructor Documentation

25.61.2.1 gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (const char * *name* = " ", VR const & *vr* = VR::INVALID, VM const & *vm* = VM::VM0, const char * *desc* = " ") [inline]

25.61.3 Member Function Documentation

25.61.3.1 const char* gdcm::CSAHeaderDictEntry::GetDescription ( ) const [inline]

Set/Get Description.

25.61.3.2 const char* gdcm::CSAHeaderDictEntry::GetName ( ) const [inline]

Set/Get Name.

Referenced by operator<().

25.61.3.3 const VM& gdcm::CSAHeaderDictEntry::GetVM ( ) const [inline]

Set/Get VM.

25.61.3.4 const VR& gdcm::CSAHeaderDictEntry::GetVR ( ) const [inline]

Set/Get VR.

25.61.3.5 bool gdcm::CSAHeaderDictEntry::operator< (const CSAHeaderDictEntry & *entry*) const [inline]

References GetName().

25.61.3.6 void gdcm::CSAHeaderDictEntry::SetDescription (const char * *desc*) [inline]

25.61.3.7 void gdcm::CSAHeaderDictEntry::SetName (const char * *name*) [inline]

25.61.3.8 void gdcm::CSAHeaderDictEntry::SetVM (VM const & *vm*) [inline]

25.61.3.9 void gdcm::CSAHeaderDictEntry::SetVR (const VR & *vr*) [inline]

25.61.4 Friends And Related Function Documentation

25.61.4.1 std::ostream& operator<< (std::ostream & *_os,* const CSAHeaderDictEntry & *_val*) [friend]

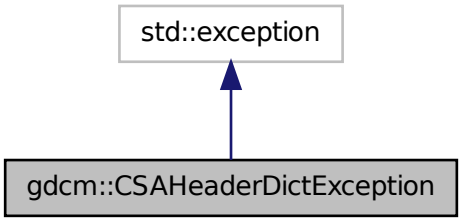
The documentation for this class was generated from the following file:

- gdcmCSAHeaderDictEntry.h

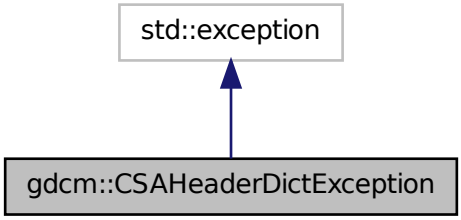
25.62 gdcm::CSAHeaderDictException Class Reference

```
#include <gdcmCSAHeaderDict.h>
```

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for gdcm::CSAHeaderDictException:



The documentation for this class was generated from the following file:

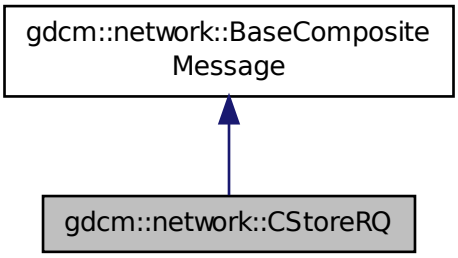
- gdcmCSAHeaderDict.h

25.63 gdcm::network::CStoreRQ Class Reference

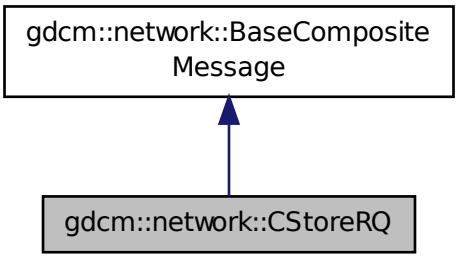
CStoreRQ this file defines the messages for the cecho action.

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for gdcm::network::CStoreRQ:



Collaboration diagram for gdcm::network::CStoreRQ:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const File &file)

25.63.1 Detailed Description

CStoreRQ this file defines the messages for the cecho action.

25.63.2 Member Function Documentation

25.63.2.1 std::vector<PresentationDataValue> gdcm::network::CStoreRQ::ConstructPDV (const ULConnection & *inConnection,* const File & *file*)

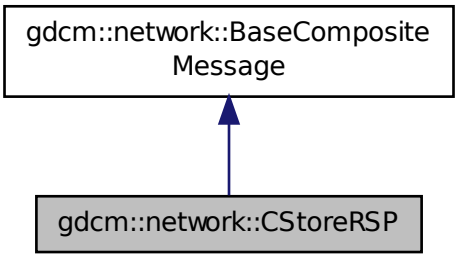
The documentation for this class was generated from the following file:

- gdcmCStoreMessages.h

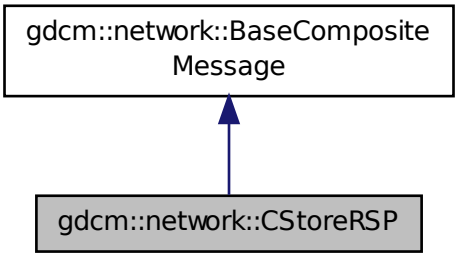
25.64 gdcm::network::CStoreRSP Class Reference

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for gdcm::network::CStoreRSP:



Collaboration diagram for gdcm::network::CStoreRSP:



Public Member Functions

- std::vector
  < PresentationDataValue > ConstructPDV (const DataSet ∗inDataSet, const BasePDU ∗inPC)

25.64.1 Member Function Documentation

25.64.1.1 std::vector<PresentationDataValue> gdcm::network::CStoreRSP::ConstructPDV (const DataSet ∗ *inDataSet,* const BasePDU ∗ *inPC*)

The documentation for this class was generated from the following file:

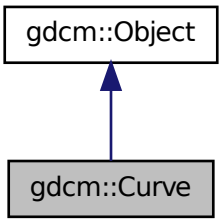
- gdcmCStoreMessages.h

25.65 gdcm::Curve Class Reference

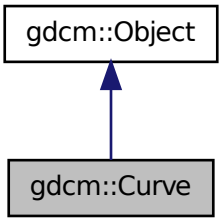
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

```
#include <gdcmCurve.h>
```

Inheritance diagram for gdcm::Curve:



Collaboration diagram for gdcm::Curve:



Public Member Functions

- Curve ()
- Curve (Curve const &ov)
- ~Curve ()
- void Decode (std::istream &is, std::ostream &os)
- void GetAsPoints (float *array) const
- unsigned short GetDataValueRepresentation () const
- unsigned short GetDimensions () const
- unsigned short GetGroup () const
- unsigned short GetNumberOfPoints () const

- const char ∗ GetTypeOfData () const
- const char ∗ GetTypeOfDataDescription () const
- bool IsEmpty () const
- void Print (std::ostream &) const
- void SetCoordinateStartValue (unsigned short v)
- void SetCoordinateStepValue (unsigned short v)
- void SetCurve (const char ∗array, unsigned int length)
- void SetCurveDataDescriptor (const uint16_t ∗values, size_t num)
- void SetCurveDescription (const char ∗curvedescription)
- void SetDataValueRepresentation (unsigned short datavaluerepresentation)
- void SetDimensions (unsigned short dimensions)
- void SetGroup (unsigned short group)
- void SetNumberOfPoints (unsigned short numberofpoints)
- void SetTypeOfData (const char ∗typeofdata)
- void Update (const DataElement &de)

Static Public Member Functions

- static unsigned int GetNumberOfCurves (DataSet const &ds)

Additional Inherited Members

25.65.1 Detailed Description

Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

Examples:

- GE_DLX-8-MONO2-Multiframe-Jpeg_Lossless.dcm
- GE_DLX-8-MONO2-Multiframe.dcm
- gdcmSampleData/Philips_Medical_Images/integris_HV_5000/xa_integris.dcm
- TOSHIBA-CurveData[1-3].dcm

25.65.2 Constructor & Destructor Documentation

25.65.2.1 gdcm::Curve::Curve ( )

25.65.2.2 gdcm::Curve::∼Curve ( )

25.65.2.3 gdcm::Curve::Curve (Curve const & *ov*)

25.65.3 Member Function Documentation

25.65.3.1 void gdcm::Curve::Decode (std::istream & *is*, std::ostream & *os*)

25.65.3.2 void gdcm::Curve::GetAsPoints (float ∗ *array*) const

25.65.3.3 unsigned short gdcm::Curve::GetDataValueRepresentation ( ) const

25.65.3.4 unsigned short gdcm::Curve::GetDimensions ( ) const

25.65.3.5 unsigned short gdcm::Curve::GetGroup ( ) const

25.65.3.6 static unsigned int gdcm::Curve::GetNumberOfCurves (DataSet const & *ds*) [static]

25.65.3.7 unsigned short gdcm::Curve::GetNumberOfPoints ( ) const

25.65.3.8 const char* gdcm::Curve::GetTypeOfData ( ) const

25.65.3.9 const char* gdcm::Curve::GetTypeOfDataDescription ( ) const

25.65.3.10 bool gdcm::Curve::IsEmpty ( ) const

25.65.3.11 void gdcm::Curve::Print (std::ostream &) const [virtual]

Reimplemented from gdcm::Object.

25.65.3.12 void gdcm::Curve::SetCoordinateStartValue (unsigned short *v*)

25.65.3.13 void gdcm::Curve::SetCoordinateStepValue (unsigned short *v*)

25.65.3.14 void gdcm::Curve::SetCurve (const char * *array,* unsigned int *length*)

25.65.3.15 void gdcm::Curve::SetCurveDataDescriptor (const uint16_t * *values,* size_t *num*)

25.65.3.16 void gdcm::Curve::SetCurveDescription (const char * *curvedescription*)

25.65.3.17 void gdcm::Curve::SetDataValueRepresentation (unsigned short *datavaluerepresentation*)

25.65.3.18 void gdcm::Curve::SetDimensions (unsigned short *dimensions*)

25.65.3.19 void gdcm::Curve::SetGroup (unsigned short *group*)

25.65.3.20 void gdcm::Curve::SetNumberOfPoints (unsigned short *numberofpoints*)

25.65.3.21 void gdcm::Curve::SetTypeOfData (const char * *typeofdata*)

25.65.3.22 void gdcm::Curve::Update (const DataElement & *de*)

The documentation for this class was generated from the following file:

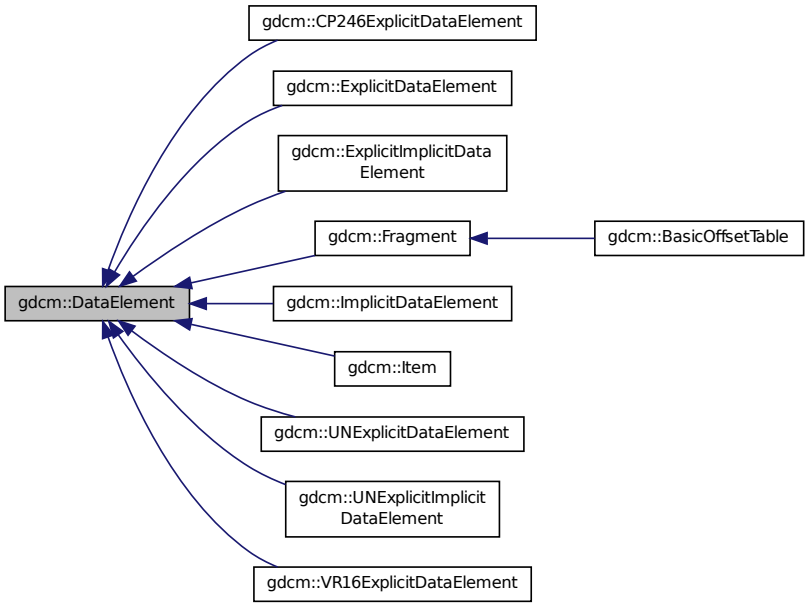
- gdcmCurve.h

25.66 gdcm::DataElement Class Reference

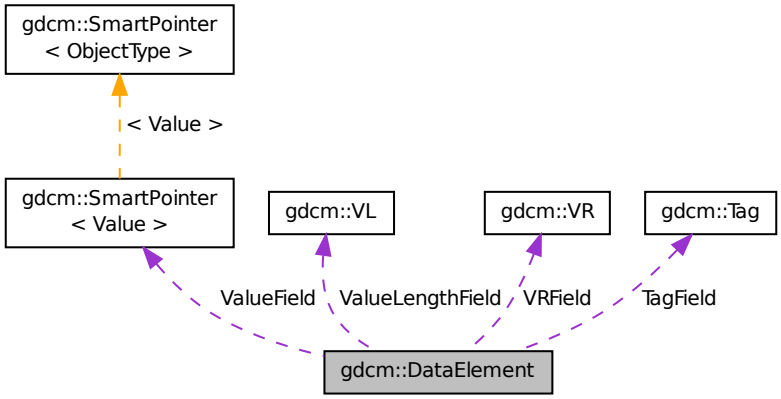
Class to represent a Data Element either Implicit or Explicit.

```
#include <gdcmDataElement.h>
```

Inheritance diagram for gdcm::DataElement:



Collaboration diagram for gdcm::DataElement:



Public Member Functions

- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- DataElement (const DataElement &_val)
- void Clear ()

Clear Data Element (make Value empty and invalidate Tag & VR)

- void Empty ()

  Make Data Element empty (no Value)
- const ByteValue * GetByteValue () const
- template<typename TDE >
  VL GetLength () const
- const SequenceOfFragments * GetSequenceOfFragments () const
- const SequenceOfItems * GetSequenceOfItems () const
- SequenceOfItems * GetSequenceOfItems ()
- const Tag & GetTag () const

  Get Tag.
- Tag & GetTag ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- Value & GetValue ()
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- const VL & GetVL () const

  Get VL.
- VL & GetVL ()
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &de)
- bool operator== (const DataElement &de) const
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TDE , typename TSwap >
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE , typename TSwap >
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE , typename TSwap >
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE , typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE , typename TSwap >
  const std::ostream & Write (std::ostream &os) const

Protected Types

- typedef SmartPointer< Value > ValuePtr

Protected Attributes

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

Friends

- std::ostream & operator<< (std::ostream &_os, const DataElement &_val)

25.66.1 Detailed Description

Class to represent a Data Element either Implicit or Explicit.

DATA ELEMENT: A unit of information as defined by a single entry in the data dictionary. An encoded Information Object Definition (IOD) Attribute that is composed of, at a minimum, three fields: a Data Element Tag, a Value Length, and a Value Field. For some specific Transfer Syntaxes, a Data Element also contains a VR Field where the Value Representation of that Data Element is specified explicitly.

Design:

- A DataElement in GDCM always store VL (Value Length) on a 32 bits integer even when VL is 16 bits
- A DataElement always store the VR even for Implicit TS, in which case VR is defaulted to VR::INVALID
- For Item start/end (See 0xfffe tags), Value is NULL

See also

ExplicitDataElement ImplicitDataElement

Examples:

ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

25.66.2 Member Typedef Documentation

25.66.2.1 typedef SmartPointer<Value> gdcm::DataElement::ValuePtr [protected]

25.66.3 Constructor & Destructor Documentation

25.66.3.1 gdcm::DataElement::DataElement (const Tag & *t* = Tag (0), const VL & *vl* = 0, const VR & *vr* = VR::INVALID) [inline]

25.66.3.2 gdcm::DataElement::DataElement (const DataElement & _val) [inline]

25.66.4 Member Function Documentation

25.66.4.1 void gdcm::DataElement::Clear ( ) [inline]

Clear Data Element (make Value empty and invalidate Tag & VR)

Reimplemented in gdcm::Item.

References gdcm::VR::INVALID.

Referenced by gdcm::Item::Clear().

25.66.4.2 void gdcm::DataElement::Empty ( ) [inline]

Make Data Element empty (no Value)

25.66.4.3 const ByteValue* gdcm::DataElement::GetByteValue ( ) const [inline]

Return the Value of DataElement as a ByteValue (if possible)

Warning

: You need to check for NULL return value

Examples:

DumpADAC.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

25.66.4.4 template<typename TDE > VL gdcm::DataElement::GetLength ( ) const [inline]

Reimplemented in gdcm::Item, gdcm::Fragment, gdcm::UNExplicitImplicitDataElement, gdcm::ExplicitImplicitDataElement, gdcm::VR16ExplicitDataElement, gdcm::CP246ExplicitDataElement, gdcm::ImplicitDataElement, gdcm::UNExplicitDataElement, and gdcm::ExplicitDataElement.

25.66.4.5 const SequenceOfFragments* gdcm::DataElement::GetSequenceOfFragments ( ) const

Return the Value of DataElement as a Sequence Of Fragments (if possible)

Warning

: You need to check for NULL return value

Examples:

FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and GetJPEGSamplePrecision.cxx.

25.66.4.6 const SequenceOfItems* gdcm::DataElement::GetSequenceOfItems ( ) const

Return the Value of DataElement as a Sequence Of Items (if possible)

Warning

: You need to check for NULL return value
: In some case a Value could not have been recognized as a SequenceOfItems in those case the return of the function will be NULL, while the Value would be a valid SequenceOfItems, in those case prefer GetValueAsSQ. In which case the code internally trigger an assert to warn developper. When in doubt do not use this function and prefer GetValueAsSQ()

Deprecated Replaced by DataElement::GetValueAsSQ() as of GDCM 2.2.

25.66.4.7 SequenceOfItems* gdcm::DataElement::GetSequenceOfItems ( )

25.66.4.8 const Tag& gdcm::DataElement::GetTag ( ) const [inline]

Get Tag.

Examples:

DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::CommandDataSet::Insert(), gdcm::FileMetaInformation::Insert(), gdcm::DataSet::Insert(), operator<(), gdcm::SequenceOfFragments::Read(), gdcm::SequenceOfItems::Read(), gdcm::CommandDataSet::-Replace(), gdcm::FileMetaInformation::Replace(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

25.66.4.9 Tag& gdcm::DataElement::GetTag ( ) [inline]

25.66.4.10 Value const& gdcm::DataElement::GetValue ( ) const [inline]

Set/Get Value (bytes array, SQ of items, SQ of fragments):

Examples:

ReadAndDumpDICOMDIR.cxx.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

25.66.4.11 Value& gdcm::DataElement::GetValue ( ) [inline]

25.66.4.12 SmartPointer<SequenceOfItems> gdcm::DataElement::GetValueAsSQ ( ) const

Interpret the Value stored in the DataElement. This is more robust (but also more expensive) to call this function rather than the simpliest form: GetSequenceOfItems() It also return NULL when the Value is NOT of type SequenceOfItems

Warning

in case GetSequenceOfItems() succeed the function return this value, otherwise it creates a new SequenceOfItems, you should handle that in your case, for instance: SmartPointer<SequenceOfItems> sqi = de.GetValueAsSQ();

Examples:

ChangeSequenceUltrasound.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

25.66.4.13 const VL& gdcm::DataElement::GetVL ( ) const [inline]

Get VL.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::SequenceOfFragments::Read(), and gdcm::SequenceOfItems::Read().

25.66.4.14 VL& gdcm::DataElement::GetVL ( ) [inline]

25.66.4.15 VR const& gdcm::DataElement::GetVR ( ) const [inline]

Get VR do not set VR::SQ on bytevalue data element

Examples:

DuplicatePCDE.cxx, and GenFakeIdentifyFile.cxx.

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

25.66.4.16 bool gdcm::DataElement::IsEmpty ( ) const [inline]

Check if Data Element is empty.

Examples:

DumpADAC.cxx, DumpGEMSMovieGroup.cxx, ELSCINT1WaveToText.cxx, FixJAIBugJPEGLS.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet().

25.66.4.17 bool gdcm::DataElement::IsUndefinedLength ( ) const [inline]

return if Value Length if of undefined length

25.66.4.18 bool gdcm::DataElement::operator< (const DataElement & *de*) const [inline]

References GetTag().

25.66.4.19 DataElement& gdcm::DataElement::operator= (const DataElement & *de*) [inline]

References TagField, ValueField, ValueLengthField, and VRField.

25.66.4.20 bool gdcm::DataElement::operator== (const DataElement & *de*) const [inline]

References TagField, ValueField, ValueLengthField, and VRField.

25.66.4.21 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::Read (std::istream & *is*) [inline]

Reimplemented in gdcm::Item, gdcm::Fragment, gdcm::BasicOffsetTable, gdcm::UNExplicitImplicitDataElement, gdcm::ExplicitImplicitDataElement, gdcm::VR16ExplicitDataElement, gdcm::CP246ExplicitDataElement, gdcm::ImplicitDataElement, gdcm::UNExplicitDataElement, and gdcm::ExplicitDataElement.

25.66.4.22 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadOrSkip (std::istream & *is,* std::set< Tag > const & *skiptags*) [inline]

25.66.4.23 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadPreValue (std::istream & *is,* std::set< Tag > const & *skiptags*) [inline]

25.66.4.24 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadValue (std::istream & *is,* std::set< Tag > const & *skiptags*) [inline]

25.66.4.25 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadWithLength (std::istream & *is,* VL & *length*) [inline]

Reimplemented in gdcm::ExplicitImplicitDataElement, gdcm::VR16ExplicitDataElement, gdcm::CP246ExplicitDataElement, gdcm::ImplicitDataElement, gdcm::UNExplicitDataElement, and gdcm::ExplicitDataElement.

25.66.4.26 void gdcm::DataElement::SetByteValue (const char * *array,* VL *length*) [inline]

Set the byte value

Warning

user need to read DICOM standard for an understanding of:

- even padding
- \0 vs space padding By default even padding is achieved using \0 regardless of the of VR

Examples:

ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, Fake_Image_Using_Stream_-Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, and StreamImageReaderTest.cxx.

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), and gdcm::SequenceOfFragments::Read().

25.66.4.27 void gdcm::DataElement::SetTag (const Tag & *t*) [inline]

Set Tag Use with cautious (need to match Part 6)

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenFakeIdentifyFile.cxx, and GetSubSequenceData.cxx.

25.66.4.28 void gdcm::DataElement::SetValue (Value const & *vl*) [inline]

Warning

you need to set the ValueLengthField explicitely

Examples:

DuplicatePCDE.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

References gdcm::Value::GetLength().

25.66.4.29 void gdcm::DataElement::SetVL (const VL & *vl*) [inline]

Set VL Use with cautious (need to match Part 6), advanced user only

See also

SetByteValue

25.66.4.30 void gdcm::DataElement::SetVLToUndefined ( )

Examples:

Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

25.66.4.31 void gdcm::DataElement::SetVR (VR const & *vr*) [inline]

Set VR Use with cautious (need to match Part 6), advanced user only

Precondition

vr is a VR::VRALL (not a dual one such as OB_OW)

Examples:

Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, and StreamImageReaderTest.cxx.

References gdcm::VR::IsVRFile().

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement().

25.66.4.32 template<typename TDE , typename TSwap > const std::ostream& gdcm::DataElement::Write (std::ostream & *os*) const [inline]

Reimplemented in gdcm::Item, gdcm::Fragment, gdcm::ImplicitDataElement, and gdcm::ExplicitDataElement.

25.66.5 Friends And Related Function Documentation

25.66.5.1 std::ostream& operator<< (std::ostream & *_os,* const DataElement & *_val*) [friend]

25.66.6 Member Data Documentation

25.66.6.1 Tag gdcm::DataElement::TagField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

25.66.6.2 ValuePtr gdcm::DataElement::ValueField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

25.66.6.3 VL gdcm::DataElement::ValueLengthField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

25.66.6.4 VR gdcm::DataElement::VRField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

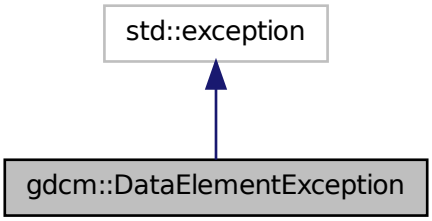
The documentation for this class was generated from the following file:

- gdcmDataElement.h

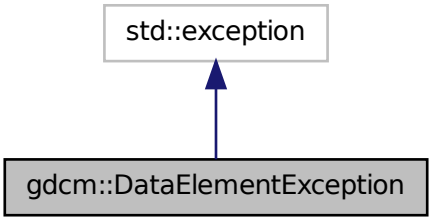
25.67 gdcm::DataElementException Class Reference

```
#include <gdcmDataSet.h>
```

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



The documentation for this class was generated from the following file:

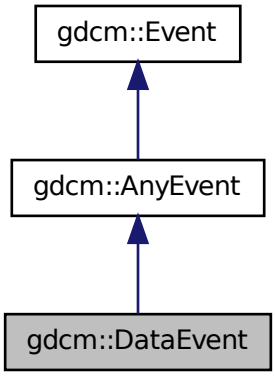
- gdcmDataSet.h

25.68 gdcm::DataEvent Class Reference

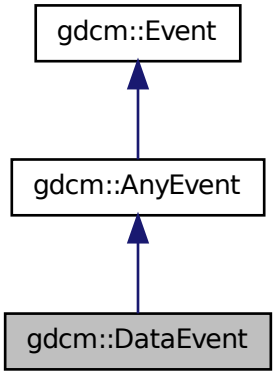
DataEvent.

```
#include <gdcmDataEvent.h>
```

Inheritance diagram for gdcm::DataEvent:



Collaboration diagram for gdcm::DataEvent:



Public Types

- typedef DataEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- DataEvent (const char ∗bytes=0, size_t len=0)

- DataEvent (const Self &s)
- virtual ∼DataEvent ()
- virtual bool CheckEvent (const ::gdcm::Event ∗e) const
- const char ∗ GetData () const
- size_t GetDataLength () const
- virtual const char ∗ GetEventName () const
- virtual ::gdcm::Event ∗ MakeObject () const
- void SetData (const char ∗bytes, size_t len)

25.68.1 Detailed Description

DataEvent.

25.68.2 Member Typedef Documentation

25.68.2.1 typedef DataEvent gdcm::DataEvent::Self

25.68.2.2 typedef AnyEvent gdcm::DataEvent::Superclass

25.68.3 Constructor & Destructor Documentation

25.68.3.1 gdcm::DataEvent::DataEvent (const char ∗ *bytes* = 0, size_t *len* = 0) [inline]

25.68.3.2 virtual gdcm::DataEvent::∼DataEvent ( ) [inline], [virtual]

25.68.3.3 gdcm::DataEvent::DataEvent (const Self & *s*) [inline]

25.68.4 Member Function Documentation

25.68.4.1 virtual bool gdcm::DataEvent::CheckEvent (const ::gdcm::Event ∗ *e*) const [inline], [virtual]

25.68.4.2 const char∗ gdcm::DataEvent::GetData ( ) const [inline]

25.68.4.3 size_t gdcm::DataEvent::GetDataLength ( ) const [inline]

25.68.4.4 virtual const char∗ gdcm::DataEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

25.68.4.5 virtual ::gdcm::Event∗ gdcm::DataEvent::MakeObject ( ) const [inline], [virtual]

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

25.68.4.6 void gdcm::DataEvent::SetData (const char ∗ *bytes*, size_t *len*) [inline]

The documentation for this class was generated from the following file:

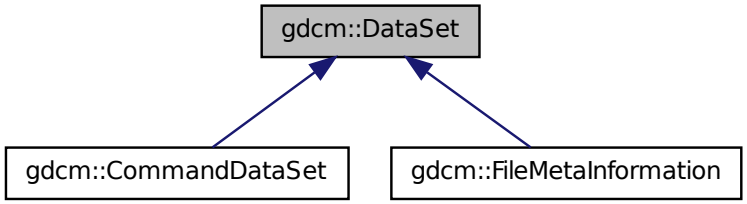
- gdcmDataEvent.h

25.69 gdcm::DataSet Class Reference

Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object.

`#include <gdcmDataSet.h>`

Inheritance diagram for gdcm::DataSet:



Public Types

- typedef DataElementSet::const_iterator ConstIterator
- typedef std::set< DataElement > DataElementSet
- typedef DataElementSet::iterator Iterator
- typedef DataElementSet::size_type SizeType

Public Member Functions

- ConstIterator Begin () const
- Iterator Begin ()
- void Clear ()
- template<typename TDE > unsigned int ComputeGroupLength (Tag const &tag) const
- ConstIterator End () const
- Iterator End ()
- bool FindDataElement (const PrivateTag &t) const

  Look up if private tag 't' is present in the dataset:
- bool FindDataElement (const Tag &t) const
- const DataElement & FindNextDataElement (const Tag &t) const
- const DataElement & GetDataElement (const Tag &t) const
- const DataElement & GetDataElement (const PrivateTag &t) const

  Return the dataelement.
- const DataElementSet & GetDES () const

- DataElementSet & GetDES ()
- template<typename TDE >
  VL GetLength () const
- std::string GetPrivateCreator (const Tag &t) const

  Return the private creator of the private tag 't':
- void Insert (const DataElement &de)
- bool IsEmpty () const

  Returns if the dataset is empty.
- const DataElement & operator() (uint16_t group, uint16_t element) const
- DataSet & operator= (DataSet const &val)
- const DataElement & operator[] (const Tag &t) const
- void Print (std::ostream &os, std::string const &indent="") const
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TDE , typename TSwap >
  std::istream & ReadNested (std::istream &is)
- template<typename TDE , typename TSwap >
  std::istream & ReadSelectedTags (std::istream &is, const std::set< Tag > &tags)
- template<typename TDE , typename TSwap >
  std::istream & ReadSelectedTagsWithLength (std::istream &is, const std::set< Tag > &tags, VL &length)
- template<typename TDE , typename TSwap >
  std::istream & ReadUpToTag (std::istream &is, const Tag &t, std::set< Tag > const &skiptags)
- template<typename TDE , typename TSwap >
  std::istream & ReadUpToTagWithLength (std::istream &is, const Tag &t, VL &length)
- template<typename TDE , typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)
- SizeType Remove (const Tag &tag)

  Completely remove a dataelement from the dataset.
- void Replace (const DataElement &de)

  Replace a dataelement with another one.
- void ReplaceEmpty (const DataElement &de)

  Only replace a DICOM attribute when it is missing or empty.
- SizeType Size () const
- template<typename TDE , typename TSwap >
  std::ostream const & Write (std::ostream &os) const

Protected Member Functions

- Tag ComputeDataElement (const PrivateTag &t) const
- const DataElement & GetDEEnd () const
- void InsertDataElement (const DataElement &de)

Friends

- class CSAHeader
- std::ostream & operator<< (std::ostream &_os, const DataSet &val)

25.69.1 Detailed Description

Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object.

Note

DATA SET: Exchanged information consisting of a structured set of Attribute values directly or indirectly related to Information Objects. The value of each Attribute in a Data Set is expressed as a Data Element. A collection of Data Elements ordered by increasing Data Element Tag number that is an encoding of the values of Attributes of a real world object.
Implementation note. If one do: DataSet ds; ds.SetLength(0); ds.Read(is); setting length to 0 actually means try to read is as if it was a root DataSet. Other value are undefined (nested dataset with undefined length) or defined length (different from 0) means nested dataset with defined length.

Warning

a DataSet does not have a Transfer Syntax type, only a File does.

Examples:

ChangeSequenceUltrasound.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, EncapsulateFileInRawData.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, SortImage.cxx, StreamImageReaderTest.cxx, and VolumeSorter.cxx.

25.69.2 Member Typedef Documentation

25.69.2.1 typedef DataElementSet::const_iterator gdcm::DataSet::ConstIterator

25.69.2.2 typedef std::set<DataElement> gdcm::DataSet::DataElementSet

25.69.2.3 typedef DataElementSet::iterator gdcm::DataSet::Iterator

25.69.2.4 typedef DataElementSet::size_type gdcm::DataSet::SizeType

25.69.3 Member Function Documentation

25.69.3.1 ConstIterator gdcm::DataSet::Begin ( ) const [inline]

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

25.69.3.2 Iterator gdcm::DataSet::Begin ( ) [inline]

25.69.3.3 void gdcm::DataSet::Clear ( ) [inline]

Referenced by gdcm::Item::Read().

25.69.3.4 Tag gdcm::DataSet::ComputeDataElement (const PrivateTag & *t*) const [protected]

25.69.3.5 template<typename TDE > unsigned int gdcm::DataSet::ComputeGroupLength (Tag const & *tag*) const [inline]

References gdcm::Tag::GetElement(), and gdcm::Tag::GetGroup().

25.69.3.6 ConstIterator gdcm::DataSet::End ( ) const [inline]

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

25.69.3.7 Iterator gdcm::DataSet::End ( ) [inline]

25.69.3.8 bool gdcm::DataSet::FindDataElement (const PrivateTag & *t*) const

Look up if private tag 't' is present in the dataset:

Examples:

ChangeSequenceUltrasound.cxx, csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet().

25.69.3.9 bool gdcm::DataSet::FindDataElement (const Tag & *t*) const [inline]

25.69.3.10 const DataElement& gdcm::DataSet::FindNextDataElement (const Tag & *t*) const [inline]

Examples:

DuplicatePCDE.cxx.

25.69.3.11 const DataElement& gdcm::DataSet::GetDataElement (const Tag & *t*) const [inline]

Return the DataElement with Tag 't'

Warning

: This only search at the 'root level' of the DataSet

Examples:

ChangeSequenceUltrasound.cxx, csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::Set(), gdcm::Attribute< Group, Element, TVR, VM::-VM1 >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet().

25.69.3.12 const DataElement& gdcm::DataSet::GetDataElement (const PrivateTag & *t*) const

Return the dataelement.

25.69.3.13 const DataElement& gdcm::DataSet::GetDEEnd ( ) const [protected]

25.69.3.14 const DataElementSet& gdcm::DataSet::GetDES ( ) const [inline]

Examples:

ReadAndDumpDICOMDIR.cxx.

25.69.3.15 DataElementSet& gdcm::DataSet::GetDES ( ) [inline]

25.69.3.16 template<typename TDE > VL gdcm::DataSet::GetLength ( ) const [inline]

25.69.3.17 std::string gdcm::DataSet::GetPrivateCreator (const Tag & *t*) const

Return the private creator of the private tag 't':

Examples:

DuplicatePCDE.cxx.

25.69.3.18 void gdcm::DataSet::Insert (const DataElement & *de*) [inline]

Insert a DataElement in the DataSet.

Warning

: Tag need to be >= 0x8 to be considered valid data element

Reimplemented in gdcm::FileMetaInformation, and gdcm::CommandDataSet.

Examples:

CreateJPIPDataSet.cxx, DuplicatePCDE.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image-_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, and StreamImageReader-Test.cxx.

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

25.69.3.19 void gdcm::DataSet::InsertDataElement (const DataElement & *de*) [inline],[protected]

References gdcmWarningMacro, gdcm::Value::GetLength(), gdcm::DataElement::GetValue(), gdcm::DataElement::-GetVL(), and gdcm::DataElement::IsEmpty().

25.69.3.20 bool gdcm::DataSet::IsEmpty ( ) const [inline]

Returns if the dataset is empty.

Referenced by gdcm::Item::Read().

25.69.3.21 const DataElement& gdcm::DataSet::operator() (uint16_t *group*, uint16_t *element*) const [inline]

25.69.3.22 DataSet& gdcm::DataSet::operator= (DataSet const & *val*) [inline]

25.69.3.23 const DataElement& gdcm::DataSet::operator[] (const Tag & *t*) const [inline]

25.69.3.24 void gdcm::DataSet::Print (std::ostream & *os*, std::string const & *indent* = " ") const [inline]

Referenced by gdcm::operator<<().

25.69.3.25 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::Read (std::istream & *is*)

Reimplemented in gdcm::FileMetaInformation, and gdcm::CommandDataSet.

25.69.3.26 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadNested (std::istream & *is*)

25.69.3.27 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTags (std::istream & *is*, const std::set< Tag > & *tags*)

25.69.3.28 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTagsWithLength (std::istream & *is*, const std::set< Tag > & *tags*, VL & *length*)

25.69.3.29 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTag (std::istream & *is*, const Tag & *t*, std::set< Tag > const & *skiptags*)

25.69.3.30 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTagWithLength (std::istream & *is*, const Tag & *t*, VL & *length*)

25.69.3.31 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadWithLength (std::istream & *is*, VL & *length*)

25.69.3.32 SizeType gdcm::DataSet::Remove (const Tag & *tag*) [inline]

Completely remove a dataelement from the dataset.

Examples:

GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

25.69.3.33 void gdcm::DataSet::Replace (const DataElement & *de*) [inline]

Replace a dataelement with another one.

Reimplemented in gdcm::FileMetaInformation, and gdcm::CommandDataSet.

Examples:

ChangeSequenceUltrasound.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeIdentifyFile.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

25.69.3.34 void gdcm::DataSet::ReplaceEmpty (const DataElement & *de*) [inline]

Only replace a DICOM attribute when it is missing or empty.

25.69.3.35 SizeType gdcm::DataSet::Size ( ) const [inline]

Examples:

DumpGEMSMovieGroup.cxx.

Referenced by gdcm::SequenceOfItems::Read().

25.69.3.36 template<typename TDE , typename TSwap > std::ostream const& gdcm::DataSet::Write (std::ostream & *os*) const

Reimplemented in gdcm::FileMetaInformation, and gdcm::CommandDataSet.

25.69.4 Friends And Related Function Documentation

25.69.4.1 friend class CSAHeader [friend]

25.69.4.2 std::ostream& operator<< (std::ostream & *_os,* const DataSet & *val*) [friend]

The documentation for this class was generated from the following file:

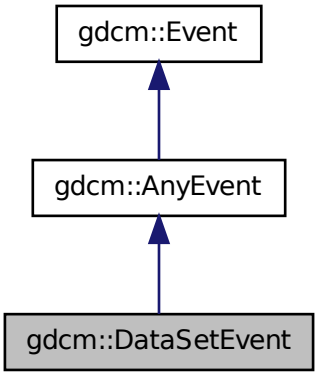
- gdcmDataSet.h

25.70 gdcm::DataSetEvent Class Reference

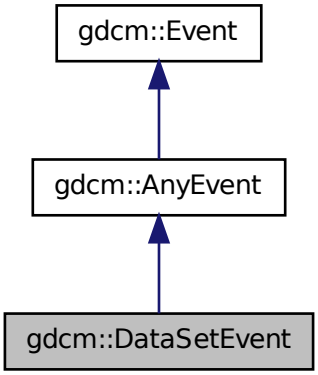
DataSetEvent Special type of event triggered during the DataSet store/move process.

```
#include <gdcmDataSetEvent.h>
```

Inheritance diagram for gdcm::DataSetEvent:



Collaboration diagram for gdcm::DataSetEvent:



Public Types

- typedef DataSetEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- DataSetEvent (DataSet const ∗ds=NULL)

- DataSetEvent (const Self &s)
- virtual ∼DataSetEvent ()
- virtual bool CheckEvent (const ::gdcm::Event ∗e) const
- DataSet const & GetDataSet () const
- virtual const char ∗ GetEventName () const
- virtual ::gdcm::Event ∗ MakeObject () const

25.70.1 Detailed Description

DataSetEvent Special type of event triggered during the DataSet store/move process.

See also

25.70.2 Member Typedef Documentation

25.70.2.1 typedef DataSetEvent gdcm::DataSetEvent::Self

25.70.2.2 typedef AnyEvent gdcm::DataSetEvent::Superclass

25.70.3 Constructor & Destructor Documentation

25.70.3.1 gdcm::DataSetEvent::DataSetEvent (DataSet const ∗ *ds* = NULL) [inline]

25.70.3.2 virtual gdcm::DataSetEvent::∼DataSetEvent ( ) [inline], [virtual]

25.70.3.3 gdcm::DataSetEvent::DataSetEvent (const Self & *s*) [inline]

25.70.4 Member Function Documentation

25.70.4.1 virtual bool gdcm::DataSetEvent::CheckEvent (const ::gdcm::Event ∗ *e*) const [inline], [virtual]

25.70.4.2 DataSet const& gdcm::DataSetEvent::GetDataSet ( ) const [inline]

25.70.4.3 virtual const char∗ gdcm::DataSetEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

25.70.4.4 virtual ::gdcm::Event∗ gdcm::DataSetEvent::MakeObject ( ) const [inline], [virtual]

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

The documentation for this class was generated from the following file:

- gdcmDataSetEvent.h

25.71 gdcm::DataSetHelper Class Reference

DataSetHelper (internal class, not intended for user level)

`#include <gdcmDataSetHelper.h>`

Static Public Member Functions

- static VR ComputeVR (File const &file, DataSet const &ds, const Tag &tag)

25.71.1 Detailed Description

DataSetHelper (internal class, not intended for user level)

25.71.2 Member Function Documentation

25.71.2.1 static VR gdcm::DataSetHelper::ComputeVR (File const & *file*, DataSet const & *ds*, const Tag & *tag*) [static]

ds -> current dataset, which is not the same as the root dataset return VR::INVALID in case of error

The documentation for this class was generated from the following file:

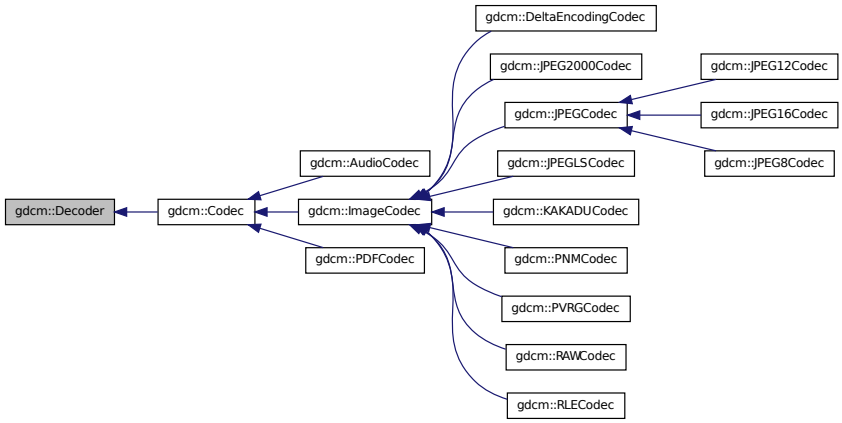
- gdcmDataSetHelper.h

25.72 gdcm::Decoder Class Reference

Decoder.

`#include <gdcmDecoder.h>`

Inheritance diagram for gdcm::Decoder:



Public Member Functions

- virtual ~Decoder ()
- virtual bool CanDecode (TransferSyntax const &) const =0

  Return whether this decoder support this transfer syntax (can decode it)
- virtual bool Decode (DataElement const &is_, DataElement &os)

  Decode.

Protected Member Functions

- virtual bool Decode (std::istream &is_, std::ostream &os)

25.72.1 Detailed Description

Decoder.

25.72.2 Constructor & Destructor Documentation

25.72.2.1 virtual gdcm::Decoder::~Decoder ( ) [inline], [virtual]

25.72.3 Member Function Documentation

25.72.3.1 virtual bool gdcm::Decoder::CanDecode (TransferSyntax const &) const [pure virtual]

Return whether this decoder support this transfer syntax (can decode it)

Implemented in gdcm::JPEGCodec, gdcm::PVRGCodec, gdcm::RLECodec, gdcm::ImageCodec, gdcm::JPEG2000Codec, gdcm::JPEGLSCodec, gdcm::PNMCodec, gdcm::RAWCodec, gdcm::AudioCodec, gdcm::PDFCodec, and gdcm::KAKADUCodec.

25.72.3.2 virtual bool gdcm::Decoder::Decode (DataElement const & *is_*, DataElement & *os*) [inline], [virtual]

Decode.

Reimplemented in gdcm::JPEGCodec, gdcm::PVRGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::ImageCodec, gdcm::DeltaEncodingCodec, gdcm::KAKADUCodec, gdcm::RAWCodec, gdcm::AudioCodec, and gdcm::PDFCodec.

25.72.3.3 virtual bool gdcm::Decoder::Decode (std::istream & *is_*, std::ostream & *os*) [inline], [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::ImageCodec, gdcm::RAWCodec, gdcm::DeltaEncodingCodec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

The documentation for this class was generated from the following file:

- gdcmDecoder.h

25.73 gdcm::DefinedTerms Class Reference

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.

```
#include <gdcmDefinedTerms.h>
```

Public Member Functions

- DefinedTerms ()

25.73.1 Detailed Description

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.

25.73.2 Constructor & Destructor Documentation

25.73.2.1 gdcm::DefinedTerms::DefinedTerms ( ) [inline]

The documentation for this class was generated from the following file:

- gdcmDefinedTerms.h

25.74 gdcm::Defs Class Reference

FIXME I do not like the name 'Defs'.

```
#include <gdcmDefs.h>
```

Public Member Functions

- Defs ()
- ~Defs ()
- const IOD & GetIODFromFile (const File &file) const
- const IODs & GetIODs () const
- IODs & GetIODs ()
- const Macros & GetMacros () const
- Macros & GetMacros ()

- const Modules & GetModules () const
- Modules & GetModules ()
- Type GetTypeFromTag (const File &file, const Tag &tag) const
- bool IsEmpty () const
- bool Verify (const File &file) const
- bool Verify (const DataSet &ds) const

Static Public Member Functions

- static const char ∗ GetIODNameFromMediaStorage (MediaStorage const &ms)

Protected Member Functions

- void LoadDefaults ()
- void LoadFromFile (const char ∗filename)

Friends

- class Global

25.74.1 Detailed Description

FIXME I do not like the name 'Defs'.

Note

bla

Examples:

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

25.74.2 Constructor & Destructor Documentation

25.74.2.1 gdcm::Defs::Defs ( )

25.74.2.2 gdcm::Defs::∼Defs ( )

25.74.3 Member Function Documentation

25.74.3.1 const IOD& gdcm::Defs::GetIODFromFile (const File & *file*) const

25.74.3.2 static const char∗ gdcm::Defs::GetIODNameFromMediaStorage (MediaStorage const & *ms*) [static]

Examples:

GenerateStandardSOPClasses.cxx.

25.74.3.3 const IODs& gdcm::Defs::GetIODs ( ) const [inline]

25.74.3.4 IODs& gdcm::Defs::GetIODs ( ) [inline]

25.74.3.5 const Macros& gdcm::Defs::GetMacros ( ) const [inline]

Users should not directly use Macro. Macro are simply a way for DICOM WG to re-use Tables. Macros are conviently wraped within Modules. See gdcm::Module API directly

25.74.3.6 Macros& gdcm::Defs::GetMacros ( ) [inline]

25.74.3.7 const Modules& gdcm::Defs::GetModules ( ) const [inline]

25.74.3.8 Modules& gdcm::Defs::GetModules ( ) [inline]

25.74.3.9 Type gdcm::Defs::GetTypeFromTag (const File & *file,* const Tag & *tag*) const

25.74.3.10 bool gdcm::Defs::IsEmpty ( ) const [inline]

25.74.3.11 void gdcm::Defs::LoadDefaults ( ) [protected]

25.74.3.12 void gdcm::Defs::LoadFromFile (const char * *filename*) [protected]

25.74.3.13 bool gdcm::Defs::Verify (const File & *file*) const

25.74.3.14 bool gdcm::Defs::Verify (const DataSet & *ds*) const

25.74.4 Friends And Related Function Documentation

25.74.4.1 friend class Global [friend]

The documentation for this class was generated from the following file:

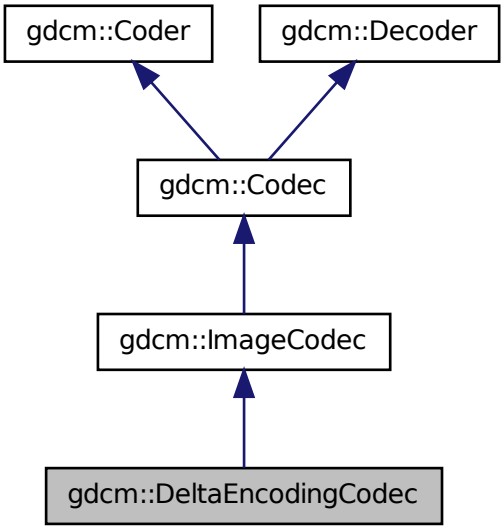
- gdcmDefs.h

25.75 gdcm::DeltaEncodingCodec Class Reference

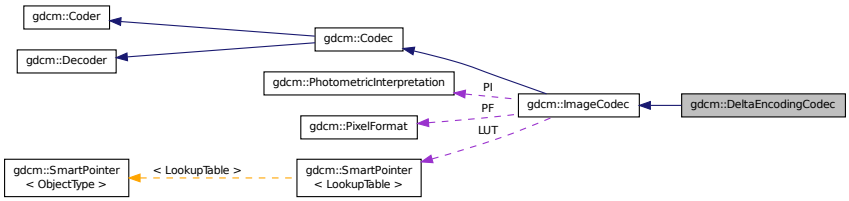
DeltaEncodingCodec compression used by some private vendor.

```
#include <gdcmDeltaEncodingCodec.h>
```

Inheritance diagram for gdcm::DeltaEncodingCodec:



Collaboration diagram for gdcm::DeltaEncodingCodec:



Public Member Functions

- DeltaEncodingCodec ()
- ~DeltaEncodingCodec ()
- bool CanDecode (TransferSyntax const &ts)
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Protected Member Functions

- bool Decode (std::istream &is, std::ostream &os)

Additional Inherited Members

25.75.1 Detailed Description

DeltaEncodingCodec compression used by some private vendor.

25.75.2 Constructor & Destructor Documentation

25.75.2.1 gdcm::DeltaEncodingCodec::DeltaEncodingCodec ( )

25.75.2.2 gdcm::DeltaEncodingCodec::~DeltaEncodingCodec ( )

25.75.3 Member Function Documentation

25.75.3.1 bool gdcm::DeltaEncodingCodec::CanDecode (TransferSyntax const & *ts*)

25.75.3.2 bool gdcm::DeltaEncodingCodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

25.75.3.3 bool gdcm::DeltaEncodingCodec::Decode (std::istream & *is*, std::ostream & *os*) [protected],[virtual]

Reimplemented from gdcm::ImageCodec.

The documentation for this class was generated from the following file:

- gdcmDeltaEncodingCodec.h

25.76 gdcm::DICOMDIR Class Reference

DICOMDIR class.

```
#include <gdcmDICOMDIR.h>
```

Public Member Functions

- DICOMDIR ()
- DICOMDIR (const FileSet &fs)

25.76.1 Detailed Description

DICOMDIR class.

Structured for handling DICOMDIR

25.76.2 Constructor & Destructor Documentation

25.76.2.1 gdcm::DICOMDIR::DICOMDIR ( ) [inline]

25.76.2.2 gdcm::DICOMDIR::DICOMDIR (const FileSet & *fs*) [inline]

The documentation for this class was generated from the following file:

- gdcmDICOMDIR.h

25.77 gdcm::DICOMDIRGenerator Class Reference

DICOMDIRGenerator class This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.

```
#include <gdcmDICOMDIRGenerator.h>
```

Public Types

- typedef Directory::FilenamesType FilenamesType
- typedef Directory::FilenameType FilenameType

Public Member Functions

- DICOMDIRGenerator ()
- ~DICOMDIRGenerator ()
- bool Generate ()

  Main function to generate the DICOMDIR.
- File & GetFile ()
- void SetDescriptor (const char *d)
- void SetFile (const File &f)

  Set/Get file. The DICOMDIR file will be valid once a call to Generate has been done.
- void SetFilenames (FilenamesType const &fns)

  Set the list of filenames from which the DICOMDIR should be generated from.
- void SetRootDirectory (FilenameType const &root)

  Set the root directory from which the filenames should be considered.

Protected Member Functions

- bool AddImageDirectoryRecord ()
- bool AddPatientDirectoryRecord ()
- bool AddSeriesDirectoryRecord ()
- bool AddStudyDirectoryRecord ()
- Scanner & GetScanner ()

25.77.1 Detailed Description

DICOMDIRGenerator class This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.

Note

PS 3.11 - 2008 / D.3.2 Physical Medium And Medium Format The STD-GEN-CD and STD-GEN-SEC-CD application profiles require the 120 mm CD-R physical medium with the ISO/IEC 9660 Media Format, as defined in PS3.12. See also PS 3.12 - 2008 / Annex F 120mm CD-R Medium (Normative) and PS 3.10 - 2008 / 8 DICOM File Service / 8.1 FILE-SET

Warning

: PS 3.11 - 2008 / D.3.1 SOP Classes and Transfer Syntaxes Composite Image & Stand-alone Storage are required to be stored as Explicit VR Little Endian Uncompressed (1.2.840.10008.1.2.1). When a DICOM file is found using another Transfer Syntax the generator will simply stops.

- Input files should be Explicit VR Little Endian
- filenames should be valid VR::CS value (16 bytes, upper case ...)

Bug : There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the gdcm::Scanner does not allow us See PS 3.11 / Table D.3-2 STD-GEN Additional DICOMDIR Keys

25.77.2 Member Typedef Documentation

25.77.2.1 typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType

25.77.2.2 typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType

25.77.3 Constructor & Destructor Documentation

25.77.3.1 gdcm::DICOMDIRGenerator::DICOMDIRGenerator ( )

25.77.3.2 gdcm::DICOMDIRGenerator::~DICOMDIRGenerator ( )

25.77.4 Member Function Documentation

25.77.4.1 bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ( ) [protected]

25.77.4.2 bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ( ) [protected]

25.77.4.3 bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ( ) [protected]

25.77.4.4 bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ( ) [protected]

25.77.4.5 bool gdcm::DICOMDIRGenerator::Generate ( )

Main function to generate the DICOMDIR.

25.77.4.6 File& gdcm::DICOMDIRGenerator::GetFile ( )

25.77.4.7 Scanner& gdcm::DICOMDIRGenerator::GetScanner ( ) [protected]

25.77.4.8 void gdcm::DICOMDIRGenerator::SetDescriptor (const char * *d*)

Set the File Set ID.

Warning

this need to be a valid VR::CS value

25.77.4.9 void gdcm::DICOMDIRGenerator::SetFile (const File & *f*)

Set/Get file. The DICOMDIR file will be valid once a call to Generate has been done.

25.77.4.10 void gdcm::DICOMDIRGenerator::SetFilenames (FilenamesType const & *fns*)

Set the list of filenames from which the DICOMDIR should be generated from.

25.77.4.11 void gdcm::DICOMDIRGenerator::SetRootDirectory (FilenameType const & *root*)

Set the root directory from which the filenames should be considered.

The documentation for this class was generated from the following file:

- gdcmDICOMDIRGenerator.h

25.78 gdcm::Dict Class Reference

Class to represent a map of DictEntry.

`#include <gdcmDict.h>`

Public Types

- typedef MapDictEntry::const_iterator ConstIterator
- typedef MapDictEntry::iterator Iterator
- typedef std::map< Tag, DictEntry > MapDictEntry

Public Member Functions

- Dict ()
- void AddDictEntry (const Tag &tag, const DictEntry &de)
- ConstIterator Begin () const
- ConstIterator End () const
- const DictEntry & GetDictEntry (const Tag &tag) const

- const DictEntry & GetDictEntryByKeyword (const char ∗keyword, Tag &tag) const
- const DictEntry & GetDictEntryByName (const char ∗name, Tag &tag) const
- const char ∗ GetKeywordFromTag (Tag const &tag) const

  Function to return the Keyword from a Tag.
- bool IsEmpty () const

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const Dict &_val)

25.78.1 Detailed Description

Class to represent a map of DictEntry.

Note

bla TODO FIXME: For Element == 0x0 need to return Name = Group Length ValueRepresentation = UL ValueMultiplicity = 1

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

25.78.2 Member Typedef Documentation

25.78.2.1 typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator

25.78.2.2 typedef MapDictEntry::iterator gdcm::Dict::Iterator

25.78.2.3 typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry

25.78.3 Constructor & Destructor Documentation

25.78.3.1 gdcm::Dict::Dict ( ) [inline]

25.78.4 Member Function Documentation

25.78.4.1 void gdcm::Dict::AddDictEntry (const Tag & *tag,* const DictEntry & *de*) [inline]

25.78.4.2 ConstIterator gdcm::Dict::Begin ( ) const [inline]

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

25.78.4.3 ConstIterator gdcm::Dict::End ( ) const [inline]

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

25.78.4.4 const DictEntry& gdcm::Dict::GetDictEntry (const Tag & *tag*) const [inline]

Examples:

GenFakeIdentifyFile.cxx, and PublicDict.cxx.

25.78.4.5 const DictEntry& gdcm::Dict::GetDictEntryByKeyword (const char * *keyword,* Tag & *tag*) const [inline]

Lookup DictEntry by keyword. Even if DICOM standard defines keyword as being unique. The lookup table is built on Tag. Therefore looking up a DictEntry by Keyword is more inefficient than looking up by Tag.

25.78.4.6 const DictEntry& gdcm::Dict::GetDictEntryByName (const char * *name,* Tag & *tag*) const [inline]

Inefficient way of looking up tag by name. Technically DICOM does not garantee uniqueness (and Curve / Overlay are there to prove it). But most of the time name is in fact uniq and can be uniquely link to a tag

Examples:

ReadAndPrintAttributes.cxx.

25.78.4.7 const char* gdcm::Dict::GetKeywordFromTag (Tag const & *tag*) const [inline]

Function to return the Keyword from a Tag.

25.78.4.8 bool gdcm::Dict::IsEmpty ( ) const [inline]

Referenced by gdcm::Dicts::IsEmpty().

25.78.4.9 void gdcm::Dict::LoadDefault ( ) [protected]

25.78.5 Friends And Related Function Documentation

25.78.5.1 friend class Dicts [friend]

25.78.5.2 std::ostream& operator<< (std::ostream & *_os,* const Dict & *_val*) [friend]

The documentation for this class was generated from the following file:

- gdcmDict.h

25.79 gdcm::DictConverter Class Reference

Class to convert a .dic file into something else:

```
#include <gdcmDictConverter.h>
```

Public Types

- enum OutputTypes {
  DICT_DEFAULT = 0,
  DICT_DEBUG,
  DICT_XML }

Public Member Functions

- DictConverter ()
- ~DictConverter ()
- void Convert ()
- const std::string & GetDictName () const
- const std::string & GetInputFilename () const
- const std::string & GetOutputFilename () const
- int GetOutputType () const
- void SetDictName (const char *name)
- void SetInputFileName (const char *filename)
- void SetOutputFileName (const char *filename)
- void SetOutputType (int type)

Static Public Member Functions

- static bool Readuint16 (const char *raw, uint16_t &ov)
- static bool ReadVM (const char *raw, VM::VMType &type)
- static bool ReadVR (const char *raw, VR::VRType &type)

Protected Member Functions

- void AddGroupLength ()
- bool ConvertToCXX (const char *raw, std::string &cxx)
- bool ConvertToXML (const char *raw, std::string &cxx)
- void WriteFooter ()
- void WriteHeader ()

25.79.1 Detailed Description

Class to convert a .dic file into something else:

- CXX code : embeded dict into shared lib (DICT_DEFAULT)
- Debug mode (DICT_DEBUG)
- XML dict (DICT_XML)

Note

25.79.2 Member Enumeration Documentation

25.79.2.1 enum gdcm::DictConverter::OutputTypes

Enumerator:

DICT_DEFAULT

DICT_DEBUG

DICT_XML

25.79.3 Constructor & Destructor Documentation

25.79.3.1 gdcm::DictConverter::DictConverter ( )

25.79.3.2 gdcm::DictConverter::~DictConverter ( )

25.79.4 Member Function Documentation

25.79.4.1 void gdcm::DictConverter::AddGroupLength ( ) [protected]

25.79.4.2 void gdcm::DictConverter::Convert ( )

25.79.4.3 bool gdcm::DictConverter::ConvertToCXX (const char * *raw*, std::string & *cxx*) [protected]

25.79.4.4 bool gdcm::DictConverter::ConvertToXML (const char * *raw*, std::string & *cxx*) [protected]

25.79.4.5 const std::string& gdcm::DictConverter::GetDictName ( ) const

25.79.4.6 const std::string& gdcm::DictConverter::GetInputFilename ( ) const

25.79.4.7 const std::string& gdcm::DictConverter::GetOutputFilename ( ) const

25.79.4.8 int gdcm::DictConverter::GetOutputType ( ) const [inline]

25.79.4.9 static bool gdcm::DictConverter::Readuint16 (const char * *raw*, uint16_t & *ov*) [static]

25.79.4.10 static bool gdcm::DictConverter::ReadVM (const char * *raw*, VM::VMType & *type*) [static]

25.79.4.11 static bool gdcm::DictConverter::ReadVR (const char * *raw*, VR::VRType & *type*) [static]

25.79.4.12 void gdcm::DictConverter::SetDictName (const char * *name*)

25.79.4.13 void gdcm::DictConverter::SetInputFileName (const char * *filename*)

25.79.4.14 void gdcm::DictConverter::SetOutputFileName (const char * *filename*)

25.79.4.15 void gdcm::DictConverter::SetOutputType (int *type*) [inline]

25.79.4.16 void gdcm::DictConverter::WriteFooter ( ) [protected]

25.79.4.17 void gdcm::DictConverter::WriteHeader ( ) [protected]

The documentation for this class was generated from the following file:

- gdcmDictConverter.h

25.80 gdcm::DictEntry Class Reference

Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information.

```
#include <gdcmDictEntry.h>
```

Public Member Functions

- DictEntry (const char ∗name="", const char ∗keyword="", VR const &vr=VR::INVALID, VM const &vm=VM::VM0, bool ret=false)
- const char ∗ GetKeyword () const

  same as GetName but without spaces...
- const char ∗ GetName () const

  Set/Get Name.
- bool GetRetired () const

  Set/Get Retired flag.
- const VM & GetVM () const

  Set/Get VM.
- const VR & GetVR () const

  Set/Get VR.
- bool IsUnique () const
- void SetElementXX (bool v)

  Set whether element is shared in multiple elements (Source Image IDs typically)
- void SetGroupXX (bool v)

  Set whether element is shared in multiple groups (Curve/Overlay typically)
- void SetKeyword (const char ∗keyword)
- void SetName (const char ∗name)
- void SetRetired (bool retired)
- void SetVM (VM const &vm)
- void SetVR (const VR &vr)

Friends

- std::ostream & operator<< (std::ostream &_os, const DictEntry &_val)

25.80.1 Detailed Description

Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information.

Note

bla TODO FIXME: Need a PublicDictEntry...indeed DictEntry has a notion of retired which does not exist in PrivateDictEntry...

See also

gdcm::Dict

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and TraverseModules.cxx.

25.80.2 Constructor & Destructor Documentation

25.80.2.1 gdcm::DictEntry::DictEntry (const char * *name* = " ", const char * *keyword* = " ", VR const & *vr* = VR::INVALID, VM const & *vm* = VM::VM0, bool *ret* = `false`) [inline]

25.80.3 Member Function Documentation

25.80.3.1 const char* gdcm::DictEntry::GetKeyword ( ) const [inline]

same as GetName but without spaces...

25.80.3.2 const char* gdcm::DictEntry::GetName ( ) const [inline]

Set/Get Name.

Referenced by gdcm::PrivateDict::PrintXML().

25.80.3.3 bool gdcm::DictEntry::GetRetired ( ) const [inline]

Set/Get Retired flag.

Examples:

GenAllVR.cxx.

25.80.3.4 const VM& gdcm::DictEntry::GetVM ( ) const [inline]

Set/Get VM.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

25.80.3.5 const VR& gdcm::DictEntry::GetVR ( ) const [inline]

Set/Get VR.

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

25.80.3.6 bool gdcm::DictEntry::IsUnique ( ) const [inline]

Return whether the name of the DataElement can be considered to be unique. As of 2008 all elements name were unique (except the expclitely 'XX' ones)

25.80.3.7 void gdcm::DictEntry::SetElementXX (bool *v*) [inline]

Set whether element is shared in multiple elements (Source Image IDs typically)

25.80.3.8 void gdcm::DictEntry::SetGroupXX (bool *v*) [inline]

Set whether element is shared in multiple groups (Curve/Overlay typically)

25.80.3.9 void gdcm::DictEntry::SetKeyword (const char * *keyword*) [inline]

25.80.3.10 void gdcm::DictEntry::SetName (const char * *name*) [inline]

25.80.3.11 void gdcm::DictEntry::SetRetired (bool *retired*) [inline]

25.80.3.12 void gdcm::DictEntry::SetVM (VM const & *vm*) [inline]

25.80.3.13 void gdcm::DictEntry::SetVR (const VR & *vr*) [inline]

Referenced by gdcm::PrivateDict::AddDictEntry().

25.80.4 Friends And Related Function Documentation

25.80.4.1 std::ostream& operator<< (std::ostream & *_os,* const DictEntry & *_val*) [friend]

The documentation for this class was generated from the following file:

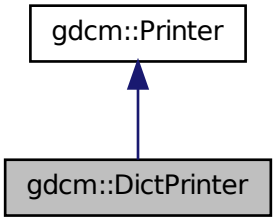
- gdcmDictEntry.h

25.81 gdcm::DictPrinter Class Reference

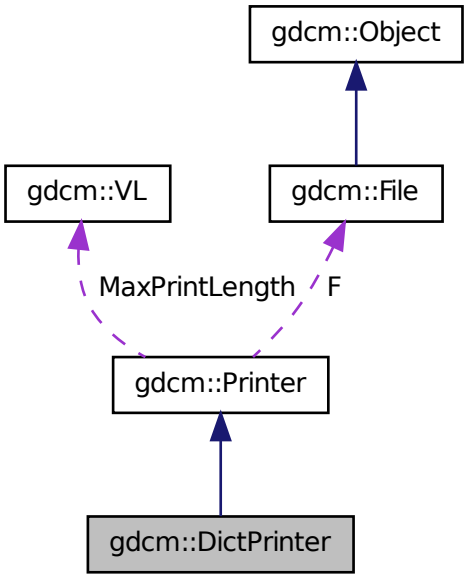
DictPrinter class.

```
#include <gdcmDictPrinter.h>
```

Inheritance diagram for gdcm::DictPrinter:



Collaboration diagram for gdcm::DictPrinter:



Public Member Functions

- DictPrinter ()
- ~DictPrinter ()
- void Print (std::ostream &os)

  Print.

Protected Member Functions

- void PrintDataElement2 (std::ostream &os, const DataSet &ds, const DataElement &ide)
- void PrintDataSet2 (std::ostream &os, const DataSet &ds)

Additional Inherited Members

25.81.1 Detailed Description

DictPrinter class.

25.81.2 Constructor & Destructor Documentation

25.81.2.1 gdcm::DictPrinter::DictPrinter ( )

25.81.2.2 gdcm::DictPrinter::∼DictPrinter ( )

25.81.3 Member Function Documentation

25.81.3.1 void gdcm::DictPrinter::Print (std::ostream & *os*)

Print.

Reimplemented from gdcm::Printer.

25.81.3.2 void gdcm::DictPrinter::PrintDataElement2 (std::ostream & *os,* const DataSet & *ds,* const DataElement & *ide*) [protected]

25.81.3.3 void gdcm::DictPrinter::PrintDataSet2 (std::ostream & *os,* const DataSet & *ds*) [protected]

The documentation for this class was generated from the following file:

- gdcmDictPrinter.h

25.82 gdcm::Dicts Class Reference

Class to manipulate the sum of knowledge (all the dict user load)

```
#include <gdcmDicts.h>
```

Public Member Functions

- Dicts ()
- ∼Dicts ()
- const CSAHeaderDict & GetCSAHeaderDict () const
- const DictEntry & GetDictEntry (const Tag &tag, const char ∗owner=NULL) const
- const DictEntry & GetDictEntry (const PrivateTag &tag) const
- const PrivateDict & GetPrivateDict () const
- PrivateDict & GetPrivateDict ()

- const Dict & GetPublicDict () const
- bool IsEmpty () const

Protected Types

- enum ConstructorType {
  PHILIPS,
  GEMS,
  SIEMENS }

Protected Member Functions

- void LoadDefaults ()

Static Protected Member Functions

- static const char ∗ GetConstructorString (ConstructorType type)

Friends

- class Global
- std::ostream & operator<< (std::ostream &_os, const Dicts &d)

25.82.1 Detailed Description

Class to manipulate the sum of knowledge (all the dict user load)

Note

bla

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

25.82.2 Member Enumeration Documentation

25.82.2.1 enum gdcm::Dicts::ConstructorType [protected]

Enumerator:

PHILIPS

GEMS

SIEMENS

25.82.3 Constructor & Destructor Documentation

25.82.3.1 gdcm::Dicts::Dicts ( )

25.82.3.2 gdcm::Dicts::~Dicts ( )

25.82.4 Member Function Documentation

25.82.4.1 static const char* gdcm::Dicts::GetConstructorString (ConstructorType *type*) [static], [protected]

25.82.4.2 const CSAHeaderDict& gdcm::Dicts::GetCSAHeaderDict ( ) const

Examples:

MrProtocol.cxx.

25.82.4.3 const DictEntry& gdcm::Dicts::GetDictEntry (const Tag & *tag,* const char * *owner* = NULL) const

works for both public and private dicts: owner is null for public dict

Warning

owner need to be set to appropriate owner for call to work. see

Examples:

PublicDict.cxx.

25.82.4.4 const DictEntry& gdcm::Dicts::GetDictEntry (const PrivateTag & *tag*) const

25.82.4.5 const PrivateDict& gdcm::Dicts::GetPrivateDict ( ) const

25.82.4.6 PrivateDict& gdcm::Dicts::GetPrivateDict ( )

25.82.4.7 const Dict& gdcm::Dicts::GetPublicDict ( ) const

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

25.82.4.8 bool gdcm::Dicts::IsEmpty ( ) const [inline]

References gdcm::Dict::IsEmpty().

25.82.4.9 void gdcm::Dicts::LoadDefaults ( ) [protected]

25.82.5 Friends And Related Function Documentation

25.82.5.1 friend class Global [friend]

25.82.5.2 std::ostream& operator<< (std::ostream & _os, const Dicts & d) [friend]

The documentation for this class was generated from the following file:

- gdcmDicts.h

25.83 gdcm::network::DIMSE Class Reference

DIMSE PS 3.7 - 2009 Annex E Command Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS Table E.1-1 COMMAND FIELDS (PART 1)

```
#include <gdcmDIMSE.h>
```

Public Types

- enum CommandTypes {
  C_STORE_RQ = 0x0001,
  C_STORE_RSP = 0x8001,
  C_GET_RQ = 0x0010,
  C_GET_RSP = 0x8010,
  C_FIND_RQ = 0x0020,
  C_FIND_RSP = 0x8020,
  C_MOVE_RQ = 0x0021,
  C_MOVE_RSP = 0x8021,
  C_ECHO_RQ = 0x0030,
  C_ECHO_RSP = 0x8030,
  N_EVENT_REPORT_RQ = 0x0100,
  N_EVENT_REPORT_RSP = 0x8100,
  N_GET_RQ = 0x0110,
  N_GET_RSP = 0x8110,
  N_SET_RQ = 0x0120,
  N_SET_RSP = 0x8120,
  N_ACTION_RQ = 0x0130,
  N_ACTION_RSP = 0x8130,
  N_CREATE_RQ = 0x0140,
  N_CREATE_RSP = 0x8140,
  N_DELETE_RQ = 0x0150,
  N_DELETE_RSP = 0x8150,
  C_CANCEL_RQ = 0x0FFF }

25.83.1 Detailed Description

DIMSE PS 3.7 - 2009 Annex E Command Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS Table E.1-1 COMMAND FIELDS (PART 1)

25.83.2 Member Enumeration Documentation

25.83.2.1 enum gdcm::network::DIMSE::CommandTypes

Enumerator:

C_STORE_RQ

C_STORE_RSP

C_GET_RQ

C_GET_RSP

C_FIND_RQ

C_FIND_RSP

C_MOVE_RQ

C_MOVE_RSP

C_ECHO_RQ

C_ECHO_RSP

N_EVENT_REPORT_RQ

N_EVENT_REPORT_RSP

N_GET_RQ

N_GET_RSP

N_SET_RQ

N_SET_RSP

N_ACTION_RQ

N_ACTION_RSP

N_CREATE_RQ

N_CREATE_RSP

N_DELETE_RQ

N_DELETE_RSP

C_CANCEL_RQ

The documentation for this class was generated from the following file:

- gdcmDIMSE.h

25.84 gdcm::DirectionCosines Class Reference

class to handle DirectionCosines

```
#include <gdcmDirectionCosines.h>
```

Public Member Functions

- DirectionCosines ()
- DirectionCosines (const double dircos[6])
- ~DirectionCosines ()
- double ComputeDistAlongNormal (const double ipp[3]) const

  Compute the distance along the normal.

- void Cross (double z[3]) const

  Compute Cross product.
- double CrossDot (DirectionCosines const &dc) const

  Compute the Dot product of the two cross vector of both DirectionCosines object.
- double Dot () const

  Compute Dot.
- bool IsValid () const

  Return whether or not this is a valid direction cosines.
- void Normalize ()

  Normalize in-place.
- operator const double ∗ () const

  Make the class behave like a const double ∗.
- void Print (std::ostream &) const

  Print.
- bool SetFromString (const char ∗str)

25.84.1 Detailed Description

class to handle DirectionCosines

Examples:

DiscriminateVolume.cxx.

25.84.2 Constructor & Destructor Documentation

25.84.2.1 gdcm::DirectionCosines::DirectionCosines ( )

25.84.2.2 gdcm::DirectionCosines::DirectionCosines (const double *dircos[6]*)

25.84.2.3 gdcm::DirectionCosines::∼DirectionCosines ( )

25.84.3 Member Function Documentation

25.84.3.1 double gdcm::DirectionCosines::ComputeDistAlongNormal (const double *ipp[3]*) const

Compute the distance along the normal.

25.84.3.2 void gdcm::DirectionCosines::Cross (double *z[3]*) const

Compute Cross product.

25.84.3.3 double gdcm::DirectionCosines::CrossDot (DirectionCosines const & *dc*) const

Compute the Dot product of the two cross vector of both DirectionCosines object.

Examples:

DiscriminateVolume.cxx.

25.84.3.4 double gdcm::DirectionCosines::Dot ( ) const

Compute Dot.

25.84.3.5 bool gdcm::DirectionCosines::IsValid ( ) const

Return whether or not this is a valid direction cosines.

25.84.3.6 void gdcm::DirectionCosines::Normalize ( )

Normalize in-place.

25.84.3.7 gdcm::DirectionCosines::operator const double * ( ) const [inline]

Make the class behave like a const double *.

25.84.3.8 void gdcm::DirectionCosines::Print (std::ostream &) const

Print.

25.84.3.9 bool gdcm::DirectionCosines::SetFromString (const char * *str*)

Initialize from string str. It requires 6 floating point separated by a backslash character.

Examples:

DiscriminateVolume.cxx.

The documentation for this class was generated from the following file:

- gdcmDirectionCosines.h

25.85 gdcm::Directory Class Reference

Class for manipulation directories.

```
#include <gdcmDirectory.h>
```

Public Types

- typedef std::vector< FilenameType > FilenamesType
- typedef std::string FilenameType

Public Member Functions

- Directory ()
- ~Directory ()

- FilenamesType const & GetDirectories () const

  Return the Directories traversed.
- FilenamesType const & GetFilenames () const

  Set/Get the file names within the directory.
- FilenameType const & GetToplevel () const

  Get the name of the toplevel directory.
- unsigned int Load (FilenameType const &name, bool recursive=false)
- void Print (std::ostream &os=std::cout) const

  Print.

Protected Member Functions

- unsigned int Explore (FilenameType const &name, bool recursive)

  Return number of file found when 'recursive'ly exploring directory `name`

Friends

- std::ostream & operator<< (std::ostream &_os, const Directory &d)

25.85.1 Detailed Description

Class for manipulation directories.

Note

This implementation provide a cross platform implementation for manipulating directores: basically traversing directories and harvesting files
will not take into account unix type hidden file recursive option will not look into UNIX type hidden directory (those starting with a '.')
Since python or C# provide there own equivalent implementation, in which case gdcm::Directory does not make much sense.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmorthoplanes.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcm.cxx, and VolumeSorter.cxx.

25.85.2 Member Typedef Documentation

25.85.2.1 typedef std::vector<FilenameType> gdcm::Directory::FilenamesType

Examples:

DiscriminateVolume.cxx.

25.85.2.2 typedef std::string gdcm::Directory::FilenameType

25.85.3 Constructor & Destructor Documentation

25.85.3.1 gdcm::Directory::Directory ( ) [inline]

25.85.3.2 gdcm::Directory::~Directory ( ) [inline]

25.85.4 Member Function Documentation

25.85.4.1 unsigned int gdcm::Directory::Explore (FilenameType const & *name*, bool *recursive*) [protected]

Return number of file found when 'recursive'ly exploring directory `name`

25.85.4.2 FilenamesType const& gdcm::Directory::GetDirectories ( ) const [inline]

Return the Directories traversed.

25.85.4.3 FilenamesType const& gdcm::Directory::GetFilenames ( ) const [inline]

Set/Get the file names within the directory.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmorthoplanes.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcm.cxx, and VolumeSorter.cxx.

25.85.4.4 FilenameType const& gdcm::Directory::GetToplevel ( ) const [inline]

Get the name of the toplevel directory.

25.85.4.5 unsigned int gdcm::Directory::Load (FilenameType const & *name*, bool *recursive* = `false`) [inline]

construct a list of filenames and subdirectory beneath directory: name

Warning

: hidden file and hidden directory are not loaded.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmorthoplanes.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcm.cxx, and VolumeSorter.cxx.

25.85.4.6 void gdcm::Directory::Print (std::ostream & *os* = `std::cout`) const

Print.

Examples:

SortImage.cxx.

Referenced by gdcm::operator<<().

25.85.5 Friends And Related Function Documentation

25.85.5.1 std::ostream& operator<< (std::ostream & *_os,* const Directory & *d*) [friend]

The documentation for this class was generated from the following file:

- gdcmDirectory.h

25.86 gdcm::DirectoryHelper Class Reference

```
#include <gdcmDirectoryHelper.h>
```

Static Public Member Functions

- static Directory::FilenamesType GetCTImageSeriesUIDs (const std::string &inDirectory)
- static Directory::FilenamesType GetFilenamesFromSeriesUIDs (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string GetFrameOfReference (const std::vector< DataSet > &inDS)
- static Directory::FilenamesType GetMRImageSeriesUIDs (const std::string &inDirectory)
- static Directory::FilenamesType GetRTStructSeriesUIDs (const std::string &inDirectory)
- static Directory::FilenamesType GetSeriesUIDsBySOPClassUID (const std::string &inDirectory, const std::string &inSOPClassUID)
- static std::string GetSOPClassUID (const std::vector< DataSet > &inDS)
- static std::vector< DataSet > LoadImageFromFiles (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string RetrieveSOPInstanceUIDFromIndex (int inIndex, const std::vector< DataSet > &inDS)
- static std::string RetrieveSOPInstanceUIDFromZPosition (double inZPos, const std::vector< DataSet > &inDS)

25.86.1 Member Function Documentation

25.86.1.1 static Directory::FilenamesType gdcm::DirectoryHelper::GetCTImageSeriesUIDs (const std::string & *inDirectory*) [static]

25.86.1.2 static Directory::FilenamesType gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs (const std::string & *inDirectory,* const std::string & *inSeriesUID*) [static]

Examples:

GenerateRTSTRUCT.cxx.

25.86.1.3 static std::string gdcm::DirectoryHelper::GetFrameOfReference (const std::vector< DataSet > & *inDS*) [static]

25.86.1.4 static Directory::FilenamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (const std::string & *inDirectory*) [static]

25.86.1.5 static Directory::FilenamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (const std::string & *inDirectory*) [static]

Examples:

GenerateRTSTRUCT.cxx.

25.86.1.6 static Directory::FilenamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (const std::string & *inDirectory,* const std::string & *inSOPClassUID*) [static]

25.86.1.7 static std::string gdcm::DirectoryHelper::GetSOPClassUID (const std::vector< DataSet > & *inDS*) [static]

25.86.1.8 static std::vector<DataSet> gdcm::DirectoryHelper::LoadImageFromFiles (const std::string & *inDirectory,* const std::string & *inSeriesUID*) [static]

25.86.1.9 static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (int *inIndex,* const std::vector< DataSet > & *inDS*) [static]

25.86.1.10 static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromZPosition (double *inZPos,* const std::vector< DataSet > & *inDS*) [static]

The documentation for this class was generated from the following file:

- gdcmDirectoryHelper.h

25.87 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

`#include <gdcmDummyValueGenerator.h>`

Static Public Member Functions

- static const char * Generate (const char *input)

25.87.1 Detailed Description

Class for generating dummy value.

See also

Anonymizer

25.87.2 Member Function Documentation

25.87.2.1 static const char* gdcm::DummyValueGenerator::Generate (const char * *input*) [static]

Generate a dummy value from an input value. This is guarantee to always return the same output value when input is identical. Return an array of bytes that can be used for anonymization purpose, return NULL on error NOT THREAD SAFE

The documentation for this class was generated from the following file:

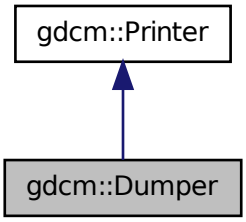
- gdcmDummyValueGenerator.h

25.88 gdcm::Dumper Class Reference

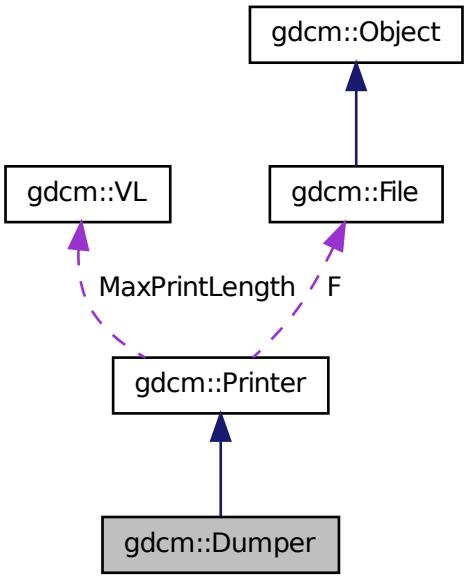
Codec class.

`#include <gdcmDumper.h>`

Inheritance diagram for gdcm::Dumper:



Collaboration diagram for gdcm::Dumper:



Public Member Functions

- Dumper ()
- ~Dumper ()

Additional Inherited Members

25.88.1 Detailed Description

Codec class.

Note

Use it to simply dump value read from the file. No interpretation is done. But it is real fast ! Almost no overhead

25.88.2 Constructor & Destructor Documentation

25.88.2.1 gdcm::Dumper::Dumper ( ) [inline]

25.88.2.2 gdcm::Dumper::~Dumper ( ) [inline]

The documentation for this class was generated from the following file:

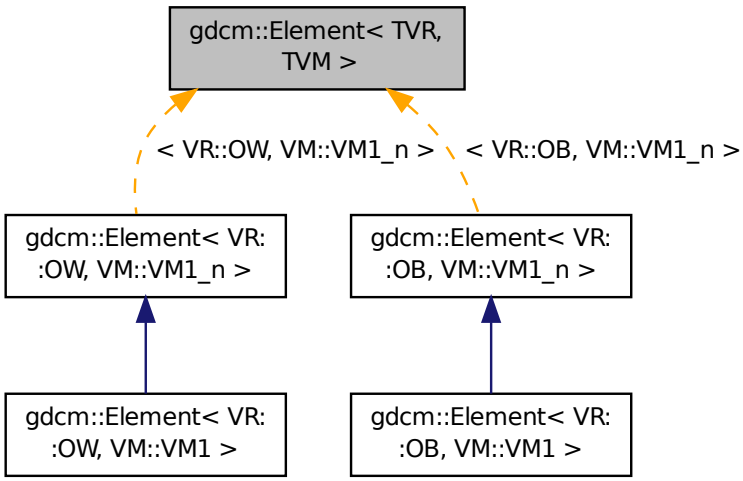
- gdcmDumper.h

25.89 gdcm::Element< TVR, TVM > Class Template Reference

Element class.

`#include <gdcmElement.h>`

Inheritance diagram for gdcm::Element< TVR, TVM >:



Public Types

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

25.89.1 Detailed Description

template<int TVR, int TVM>class gdcm::Element< TVR, TVM >

Element class.

Note

TODO

Examples:

csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_-Stream_Image_Writer.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

25.89.2 Member Typedef Documentation

25.89.2.1 template<int TVR, int TVM> typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type

25.89.3 Member Function Documentation

25.89.3.1 template<int TVR, int TVM> DataElement gdcm::Element< TVR, TVM >::GetAsDataElement ( ) const [inline]

25.89.3.2 template<int TVR, int TVM> unsigned long gdcm::Element< TVR, TVM >::GetLength ( ) const [inline]

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::-GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::Print(), gdcm::Element< VR::OB, VM::VM1_n >::Read(), gdcm::Element< TVR, VM::VM1_n >::Read(), gdcm::Element< VR::OB, VM::VM1_n >::Set(), gdcm::Element< TVR, VM::VM1_n >::Set(), gdcm::Element< VR::OB, VM::VM1_n >::SetNoSwap(), gdcm::Element< TVR, VM::VM1_n >-::SetNoSwap(), gdcm::Element< VR::OB, VM::VM1_n >::Write(), gdcm::Element< TVR, VM::VM1_n >::Write(), and gdcm::Element< TVR, VM::VM1_n >::WriteASCII().

25.89.3.3 template<int TVR, int TVM> const VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (unsigned int *idx* = 0) const [inline]

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::operator[]() , and gdcm::Element< TVR, VM::VM1_n >-::operator[]().

25.89.3.4 template<int TVR, int TVM> VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (unsigned int *idx* = 0) [inline]

25.89.3.5 template<int TVR, int TVM> const VRToType<TVR>::Type* gdcm::Element< TVR, TVM >::GetValues ( ) const [inline]

25.89.3.6 template<int TVR, int TVM> static VM gdcm::Element< TVR, TVM >::GetVM ( ) [inline], [static]

25.89.3.7 template<int TVR, int TVM> static VR gdcm::Element< TVR, TVM >::GetVR ( ) [inline], [static]

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), and gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement().

25.89.3.8 template<int TVR, int TVM> VRToType<TVR>::Type gdcm::Element< TVR, TVM >::operator[] (unsigned int *idx*) const [inline]

25.89.3.9 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Print (std::ostream & *_os*) const [inline]

25.89.3.10 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Read (std::istream & *_is*) [inline]

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::Read(), gdcm::Element< TVR, VM::VM1_n >::Read(), gdcm::Element< VR::OB, VM::VM1_n >::Set(), and gdcm::Element< TVR, VM::VM1_n >::Set().

25.89.3.11 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Set (Value const & *v*) [inline]

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

25.89.3.12 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetFromDataElement (DataElement< TVR, TVM > const & *de*) [inline]

25.89.3.13 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetNoSwap (Value const & *v*) [inline], [protected]

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

25.89.3.14 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetValue (typename VRToType< TVR >::Type *v*, unsigned int *idx* = 0) [inline]

25.89.3.15 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Write (std::ostream & *_os*) const [inline]

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< VR::OB, VM::VM1_n >::Write(), gdcm::Element< TVR, VM::VM1_n >::Write(), and gdcm::Element< TVR, VM::VM1_n >::WriteASCII().

25.89.4 Member Data Documentation

25.89.4.1 template<int TVR, int TVM> VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Element< VR::OB, VM::VM1_n >::GetValue(), gdcm::Element< VR::OB, VM::VM1_n >::GetValues(), gdcm::Element< VR::OB, VM::VM1_n >::Print(), gdcm::Element< VR::AS, VM::VM5 >::Print(), gdcm::Element< VR::OB, VM::VM1_n >::Read(), gdcm::Element< VR::OB, VM::VM1_n >::Set(), gdcm::Element< TVR, VM::VM1_n >::SetLength(), gdcm::Element< VR::OB, VM::VM1_n >::SetNoSwap(), gdcm::Element< VR::OB, VM::VM1_n >::SetValue(), gdcm::Element< VR::OB, VM::VM1_n >::Write(), and gdcm::Element< TVR, VM::VM1_n >::~Element().

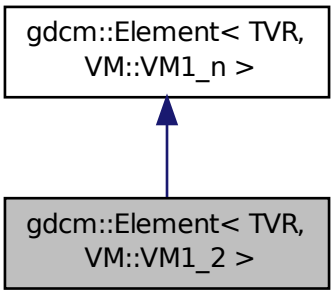
The documentation for this class was generated from the following file:

- gdcmElement.h

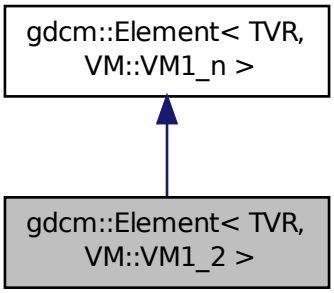
25.90 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM1_2 >:



Collaboration diagram for gdcm::Element< TVR, VM::VM1_2 >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

25.90.1 Member Typedef Documentation

25.90.1.1 template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM1_2 >::Parent

25.90.2 Member Function Documentation

25.90.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM1_2 >::SetLength (int *len*) [inline]

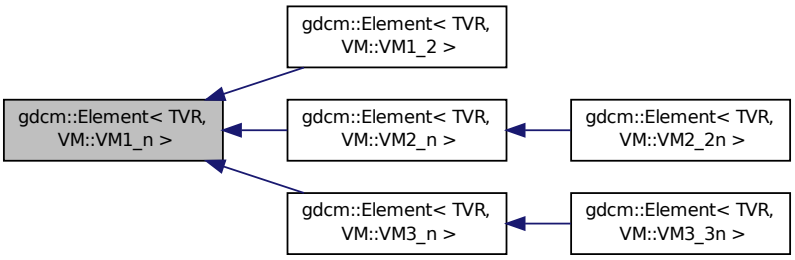
The documentation for this class was generated from the following file:

- gdcmElement.h

25.91 gdcm::Element< TVR, VM::VM1_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM1_n >:



Public Types

- typedef VRToType< TVR >::Type Type

Public Member Functions

- Element ()
- Element (const Element &_val)
- ~Element ()
- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- Element & operator= (const Element &_val)
- VRToType< TVR >::Type operator[] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetArray (const Type *array, unsigned long len, bool save=false)
- void SetFromDataElement (DataElement const &de)
- void SetLength (unsigned long len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const
- void WriteASCII (std::ostream &os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Protected Member Functions

- void SetNoSwap (Value const &v)

25.91.1 Member Typedef Documentation

25.91.1.1 template<int TVR> typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::Type

25.91.2 Constructor & Destructor Documentation

25.91.2.1 template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element ( ) [inline], [explicit]

25.91.2.2 template<int TVR> gdcm::Element< TVR, VM::VM1_n >::~Element ( ) [inline]

References gdcm::Element< TVR, TVM >::Internal.

25.91.2.3 template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element (const Element< TVR, VM::VM1_n > & *_val*) [inline]

25.91.3 Member Function Documentation

25.91.3.1 template<int TVR> DataElement gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]

References gdcm::Element< TVR, TVM >::GetLength(), gdcm::Element< TVR, TVM >::GetVR(), gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, gdcm::VR::VRASCII, and gdcm::Element< TVR, TVM >::Write().

25.91.3.2 template<int TVR> unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength ( ) const [inline]

25.91.3.3 template<int TVR> const VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) const [inline]

25.91.3.4 template<int TVR> VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) [inline]

25.91.3.5 template<int TVR> static VM gdcm::Element< TVR, VM::VM1_n >::GetVM ( ) [inline], [static]

References gdcm::VM::VM1_n.

25.91.3.6 template<int TVR> static VR gdcm::Element< TVR, VM::VM1_n >::GetVR ( ) [inline], [static]

25.91.3.7 template<int TVR> Element& gdcm::Element< TVR, VM::VM1_n >::operator= (const Element< TVR, VM::VM1_n > & *_val*) [inline]

25.91.3.8 template<int TVR> VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::operator[] (unsigned int *idx*) const [inline]

References gdcm::Element< TVR, TVM >::GetValue().

25.91.3.9 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Print (std::ostream & *_os*) const [inline]

References gdcm::Element< TVR, TVM >::GetLength().

25.91.3.10 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Read (std::istream & _is) [inline]

References gdcm::Element< TVR, TVM >::GetLength(), and gdcm::Element< TVR, TVM >::Read().

25.91.3.11 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Set (Value const & v) [inline]

References gdcm::Element< TVR, TVM >::GetLength(), gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), gdcm::Element< TVR, TVM >::Read(), and gdcm::VR::VRBINARY.

25.91.3.12 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetArray (const Type * array, unsigned long len, bool save = false) [inline]

25.91.3.13 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement (DataElement< TVR, VM::VM1_n > const & de) [inline]

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetValue(), gdcm::DataElement::GetVR(), gdcm::VR::INVALID, gdcm::Element< TVR, TVM >::Set(), gdcm::Element< TVR, TVM >::SetNoSwap(), and gdcm::VR::UN.

25.91.3.14 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetLength (unsigned long len) [inline]

References gdcm::Element< TVR, TVM >::Internal.

25.91.3.15 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetNoSwap (Value const & v) [inline], [protected]

References gdcm::Element< TVR, TVM >::GetLength(), gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), and gdcm::VR::VRBINARY.

25.91.3.16 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetValue (typename VRToType< TVR >::Type v, unsigned int idx = 0) [inline]

25.91.3.17 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Write (std::ostream & _os) const [inline]

References gdcm::Element< TVR, TVM >::GetLength(), and gdcm::Element< TVR, TVM >::Write().

25.91.3.18 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::WriteASCII (std::ostream & os) const [inline]

References gdcm::Element< TVR, TVM >::GetLength(), and gdcm::Element< TVR, TVM >::Write().

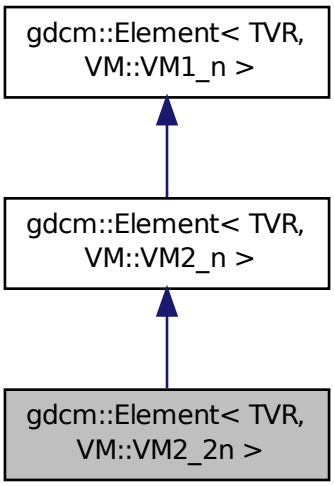
The documentation for this class was generated from the following file:

- gdcmElement.h

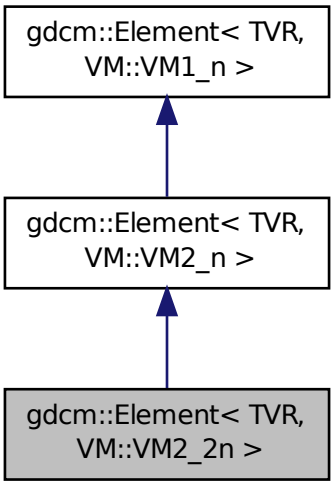
25.92 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM2_2n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM2_2n >:



Public Types

- typedef Element< TVR, VM::VM2_n > Parent

Public Member Functions

- void SetLength (int len)

25.92.1 Member Typedef Documentation

25.92.1.1 template<int TVR> typedef Element<TVR, VM::VM2_n> gdcm::Element< TVR, VM::VM2_2n >::Parent

Reimplemented from gdcm::Element< TVR, VM::VM2_n >.

25.92.2 Member Function Documentation

25.92.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM2_2n >::SetLength (int *len*) [inline]

Reimplemented from gdcm::Element< TVR, VM::VM2_n >.

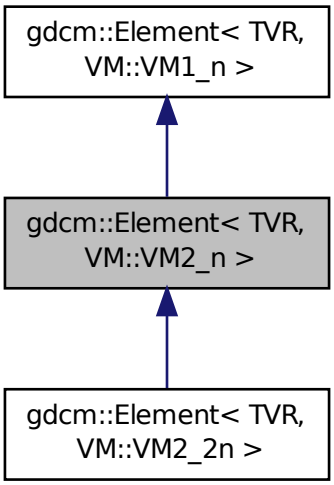
The documentation for this class was generated from the following file:

- gdcmElement.h

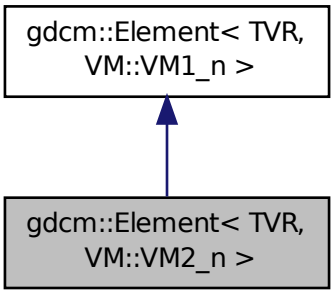
25.93 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM2_n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM2_n >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

25.93.1 Member Typedef Documentation

25.93.1.1 template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM2_n >::Parent

Reimplemented in gdcm::Element< TVR, VM::VM2_2n >.

25.93.2 Member Function Documentation

25.93.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM2_n >::SetLength (int *len*) [inline]

Reimplemented in gdcm::Element< TVR, VM::VM2_2n >.

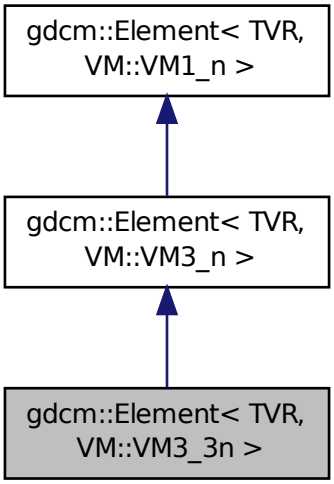
The documentation for this class was generated from the following file:

- gdcmElement.h

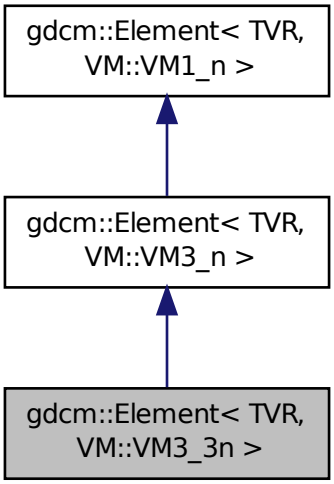
25.94 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM3_3n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_3n >:



Public Types

- typedef Element< TVR, VM::VM3_n > Parent

Public Member Functions

- void SetLength (int len)

25.94.1 Member Typedef Documentation

25.94.1.1 template<int TVR> typedef Element<TVR, VM::VM3_n> gdcm::Element< TVR, VM::VM3_3n >::Parent

Reimplemented from gdcm::Element< TVR, VM::VM3_n >.

25.94.2 Member Function Documentation

25.94.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM3_3n >::SetLength (int *len*) [inline]

Reimplemented from gdcm::Element< TVR, VM::VM3_n >.

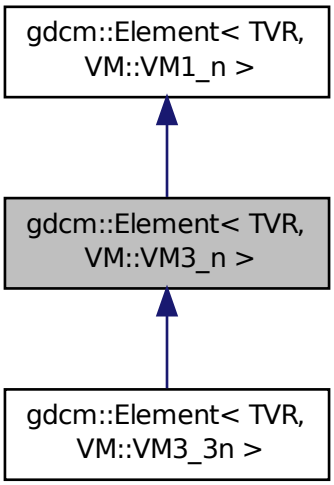
The documentation for this class was generated from the following file:

- gdcmElement.h

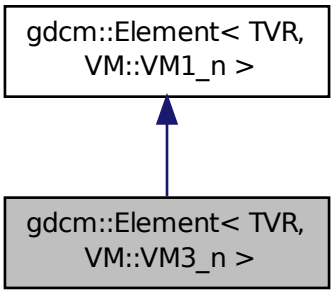
25.95 gdcm::Element< TVR, VM::VM3_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM3_n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_n >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

25.95.1 Member Typedef Documentation

25.95.1.1 template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM3_n >::Parent

Reimplemented in gdcm::Element< TVR, VM::VM3_3n >.

25.95.2 Member Function Documentation

25.95.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM3_n >::SetLength (int *len*) [inline]

Reimplemented in gdcm::Element< TVR, VM::VM3_3n >.

The documentation for this class was generated from the following file:

- gdcmElement.h

25.96 gdcm::Element< VR::AS, VM::VM5 > Class Template Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- unsigned long GetLength () const
- void Print (std::ostream &_os) const

Public Attributes

- char Internal [VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]

25.96.1 Member Function Documentation

25.96.1.1 unsigned long gdcm::Element< VR::AS, VM::VM5 >::GetLength ( ) const [inline]

25.96.1.2 void gdcm::Element< VR::AS, VM::VM5 >::Print (std::ostream & *_os*) const [inline]

References gdcm::Element< TVR, TVM >::Internal.

25.96.2 Member Data Documentation

25.96.2.1 char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]

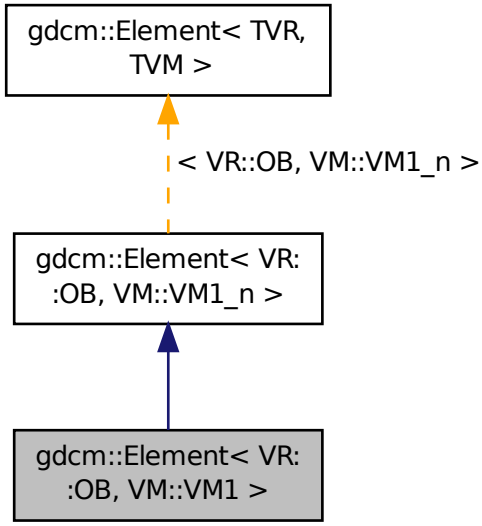
The documentation for this class was generated from the following file:

- gdcmElement.h

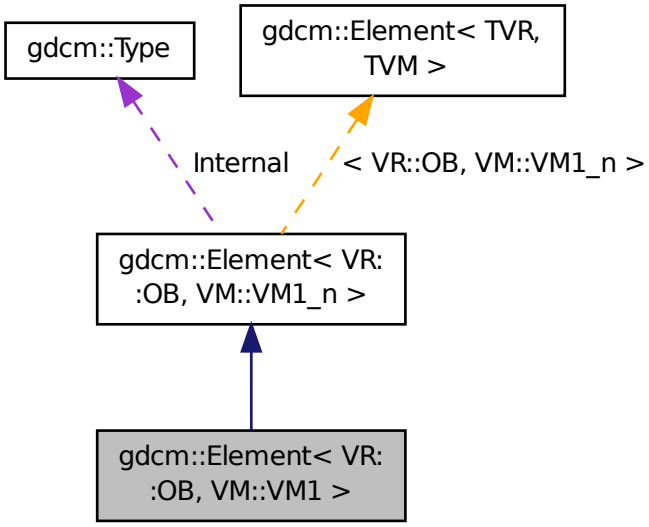
25.97 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OB, VM::VM1 >:



Additional Inherited Members

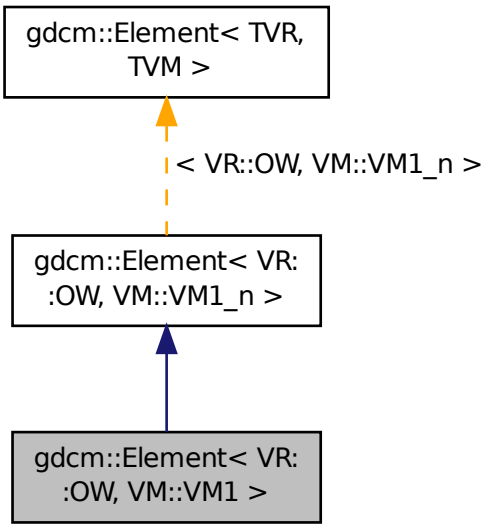
The documentation for this class was generated from the following file:

- gdcmElement.h

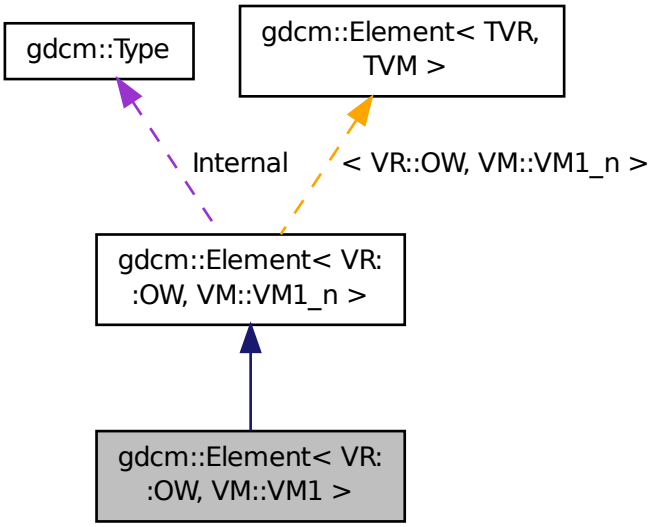
25.98 gdcm::Element< VR::OW, VM::VM1 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OW, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OW, VM::VM1 >:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmElement.h

25.99 gdcm::EncapsulatedDocument Class Reference

EncapsulatedDocument.

```
#include <gdcmEncapsulatedDocument.h>
```

Public Member Functions

- EncapsulatedDocument ()

25.99.1 Detailed Description

EncapsulatedDocument.

25.99.2 Constructor & Destructor Documentation

25.99.2.1 gdcm::EncapsulatedDocument::EncapsulatedDocument ( ) [inline]

The documentation for this class was generated from the following file:

- gdcmEncapsulatedDocument.h

25.100 gdcm::EncodingImplementation< VR::VRASCII > Class Template Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- template<>
  void Write (const float *data, unsigned long length, std::ostream &_os)
- template<>
  void Write (const double *data, unsigned long length, std::ostream &_os)

Static Public Member Functions

- template<typename T >
  static void Read (T *data, unsigned long length, std::istream &_is)
- template<typename T >
  static void ReadComputeLength (T *data, unsigned int &length, std::istream &_is)
- template<typename T >
  static void ReadNoSwap (T *data, unsigned long length, std::istream &_is)

- template<typename T >
  static void Write (const T *data, unsigned long length, std::ostream &_os)

25.100.1 Member Function Documentation

25.100.1.1 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Read (T * *data,* unsigned long *length,* std::istream & *_is*) [inline],[static]

25.100.1.2 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (T * *data,* unsigned int & *length,* std::istream & *_is*) [inline],[static]

References gdcm::backslash().

25.100.1.3 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (T * *data,* unsigned long *length,* std::istream & *_is*) [inline],[static]

25.100.1.4 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Write (const T * *data,* unsigned long *length,* std::ostream & *_os*) [inline],[static]

25.100.1.5 template<> void gdcm::EncodingImplementation< VR::VRASCII >::Write (const float * *data,* unsigned long *length,* std::ostream & *_os*) [inline]

References gdcm::to_string().

25.100.1.6 template<> void gdcm::EncodingImplementation< VR::VRASCII >::Write (const double * *data,* unsigned long *length,* std::ostream & *_os*) [inline]

References gdcm::to_string().

The documentation for this class was generated from the following file:

- gdcmElement.h

25.101 gdcm::EncodingImplementation< VR::VRBINARY > Class Template Reference

```
#include <gdcmElement.h>
```

Static Public Member Functions

- template<typename T >
  static void Read (T *data, unsigned long length, std::istream &_is)
- template<typename T >
  static void ReadComputeLength (T *data, unsigned int &length, std::istream &_is)
- template<typename T >
  static void ReadNoSwap (T *data, unsigned long length, std::istream &_is)
- template<typename T >
  static void Write (const T *data, unsigned long length, std::ostream &_os)

25.101.1 Member Function Documentation

25.101.1.1 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (T * *data,* unsigned long *length,* std::istream & *_is*) [inline],[static]

References gdcm::SwapperNoOp::SwapArray().

25.101.1.2 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (T * *data,* unsigned int & *length,* std::istream & *_is*) [inline],[static]

25.101.1.3 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (T * *data,* unsigned long *length,* std::istream & *_is*) [inline],[static]

25.101.1.4 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::Write (const T * *data,* unsigned long *length,* std::ostream & *_os*) [inline],[static]

References gdcm::SwapperNoOp::Swap().

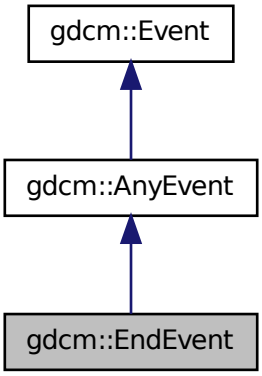
The documentation for this class was generated from the following file:

- gdcmElement.h

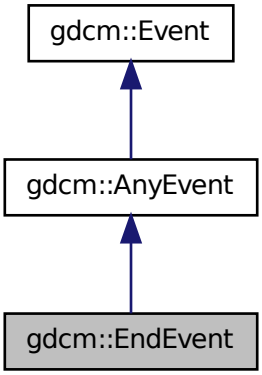
25.102 gdcm::EndEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::EndEvent:



Collaboration diagram for gdcm::EndEvent:



The documentation for this class was generated from the following file:

- gdcmEvent.h

25.103 gdcm::EnumeratedValues Class Reference

Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

```
#include <gdcmEnumeratedValues.h>
```

Public Member Functions

- EnumeratedValues ()

25.103.1 Detailed Description

Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

1. Patient Sex (0010, 0040) is an example of a Data Element having Enumerated Values. It is defined to have a Value that is either "M", "F", or "O" (see PS 3.3). No other Value shall be given to this Data Element.

2. Future modifications of this standard may add to the set of allowed values for Data Elements with Enumerated Values. Such additions by themselves may or may not require a change in SOP Class UIDs, depending on the semantics of the Data Element.

25.103.2 Constructor & Destructor Documentation

25.103.2.1 gdcm::EnumeratedValues::EnumeratedValues ( ) [inline]

The documentation for this class was generated from the following file:

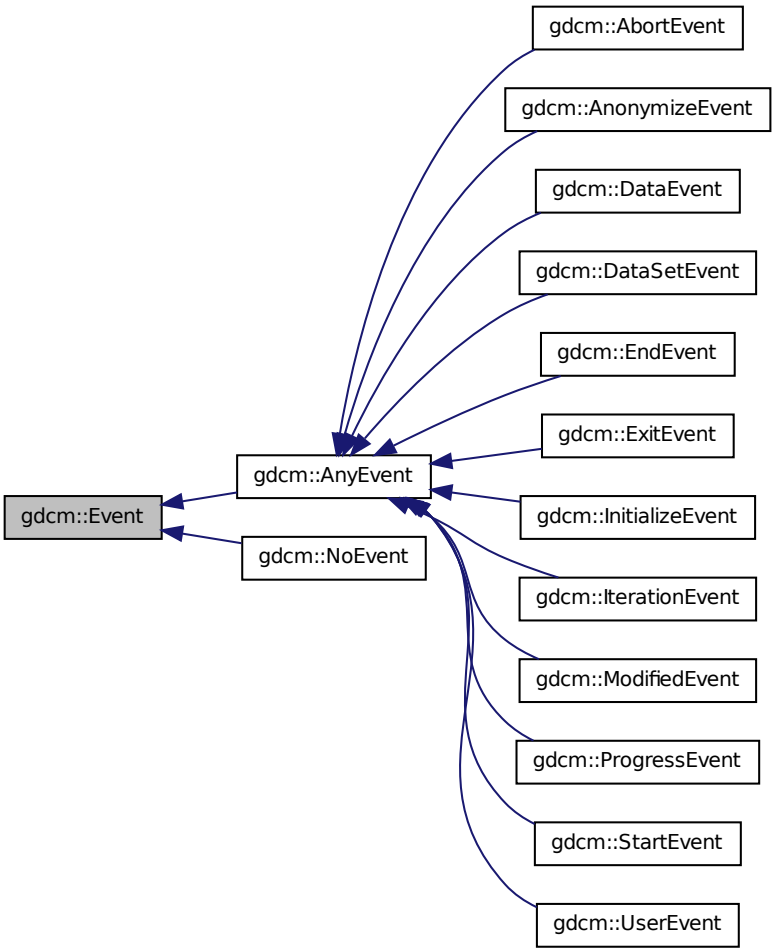
- gdcmEnumeratedValues.h

25.104 gdcm::Event Class Reference

superclass for callback/observer methods

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::Event:



Public Member Functions

- Event ()
- Event (const Event &)
- virtual ∼Event ()
- virtual bool CheckEvent (const Event ∗) const =0
- virtual const char ∗ GetEventName (void) const =0
- virtual Event ∗ MakeObject () const =0
- virtual void Print (std::ostream &os) const

25.104.1 Detailed Description

superclass for callback/observer methods

See also

Command Subject

25.104.2 Constructor & Destructor Documentation

25.104.2.1 gdcm::Event::Event ( )

25.104.2.2 gdcm::Event::Event (const Event &)

25.104.2.3 virtual gdcm::Event::∼Event ( ) [virtual]

25.104.3 Member Function Documentation

25.104.3.1 virtual bool gdcm::Event::CheckEvent (const Event ∗) const [pure virtual]

Check if given event matches or derives from this event.

25.104.3.2 virtual const char∗ gdcm::Event::GetEventName (void) const [pure virtual]

Return the StringName associated with the event.

Implemented in gdcm::ProgressEvent, gdcm::DataSetEvent, gdcm::AnonymizeEvent, and gdcm::DataEvent.

25.104.3.3 virtual Event∗ gdcm::Event::MakeObject ( ) const [pure virtual]

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implemented in gdcm::ProgressEvent, gdcm::DataSetEvent, gdcm::AnonymizeEvent, and gdcm::DataEvent.

25.104.3.4 virtual void gdcm::Event::Print (std::ostream & *os*) const [virtual]

Print Event information. This method can be overridden by specific Event subtypes. The default is to print out the type of the event.

Referenced by gdcm::operator<<().

The documentation for this class was generated from the following file:

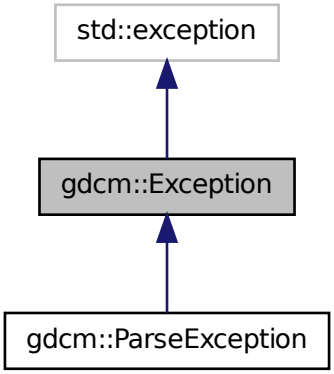
- gdcmEvent.h

25.105 gdcm::Exception Class Reference

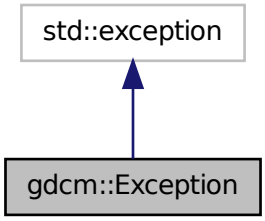
Exception.

```
#include <gdcmException.h>
```

Inheritance diagram for gdcm::Exception:



Collaboration diagram for gdcm::Exception:



Public Member Functions

- Exception (const char ∗desc="None", const char ∗file=__FILE__, unsigned int lineNumber=__LINE__, const char ∗func="")
- virtual ∼Exception () throw ()

- const char ∗ GetDescription () const

  Return the Description.
- const char ∗ what () const throw ()

  what implementation

25.105.1 Detailed Description

Exception.

Standard exception handling object.

Note

Its copy-constructor and assignment operator are generated by the compiler.

25.105.2 Constructor & Destructor Documentation

25.105.2.1 gdcm::Exception::Exception (const char ∗ *desc* = "None", const char ∗ *file* = __FILE__, unsigned int *lineNumber* = __LINE__, const char ∗ *func* = "") [inline],[explicit]

Explicit constructor, initializing the description and the text returned by what().

Note

The last parameter is ignored for the time being. It may be used to specify the function where the exception was thrown.

25.105.2.2 virtual gdcm::Exception::∼Exception ( ) throw () [inline],[virtual]

25.105.3 Member Function Documentation

25.105.3.1 const char∗ gdcm::Exception::GetDescription ( ) const [inline]

Return the Description.

Referenced by gdcm::SequenceOfItems::Read().

25.105.3.2 const char∗ gdcm::Exception::what ( ) const throw () [inline]

what implementation

Referenced by gdcm::SequenceOfFragments::Read().

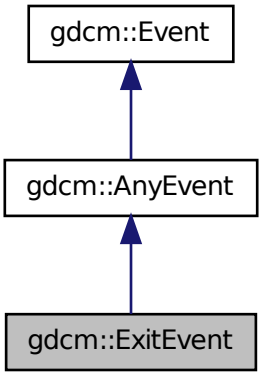
The documentation for this class was generated from the following file:

- gdcmException.h

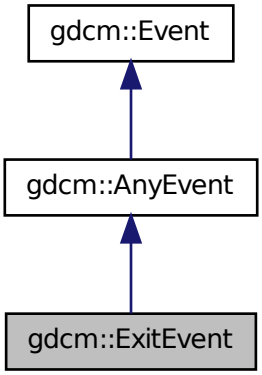
25.106 gdcm::ExitEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::ExitEvent:



Collaboration diagram for gdcm::ExitEvent:



The documentation for this class was generated from the following file:

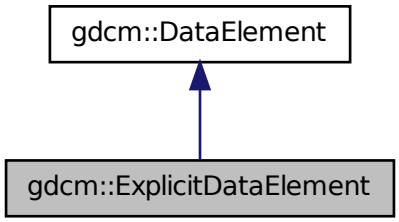
- gdcmEvent.h

25.107 gdcm::ExplicitDataElement Class Reference

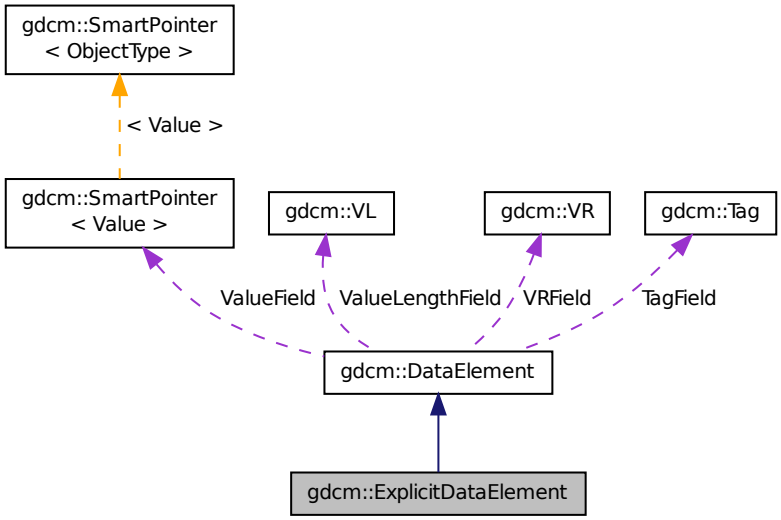
Class to read/write a DataElement as Explicit Data Element.

```
#include <gdcmExplicitDataElement.h>
```

Inheritance diagram for gdcm::ExplicitDataElement:



Collaboration diagram for gdcm::ExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)

- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

25.107.1 Detailed Description

Class to read/write a DataElement as Explicit Data Element.

Note

bla

25.107.2 Member Function Documentation

25.107.2.1 VL gdcm::ExplicitDataElement::GetLength ( ) const

Reimplemented from gdcm::DataElement.

25.107.2.2 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::Read (std::istream & *is*)

Reimplemented from gdcm::DataElement.

25.107.2.3 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadPreValue (std::istream & *is*)

25.107.2.4 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadValue (std::istream & *is*)

25.107.2.5 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*)

Reimplemented from gdcm::DataElement.

25.107.2.6 template<typename TSwap > const std::ostream& gdcm::ExplicitDataElement::Write (std::ostream & *os*) const

Reimplemented from gdcm::DataElement.

The documentation for this class was generated from the following file:

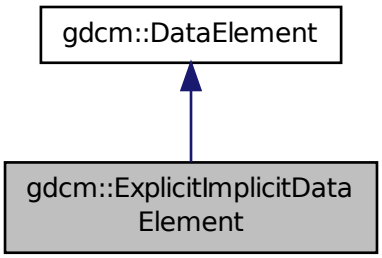
- gdcmExplicitDataElement.h

25.108 gdcm::ExplicitImplicitDataElement Class Reference

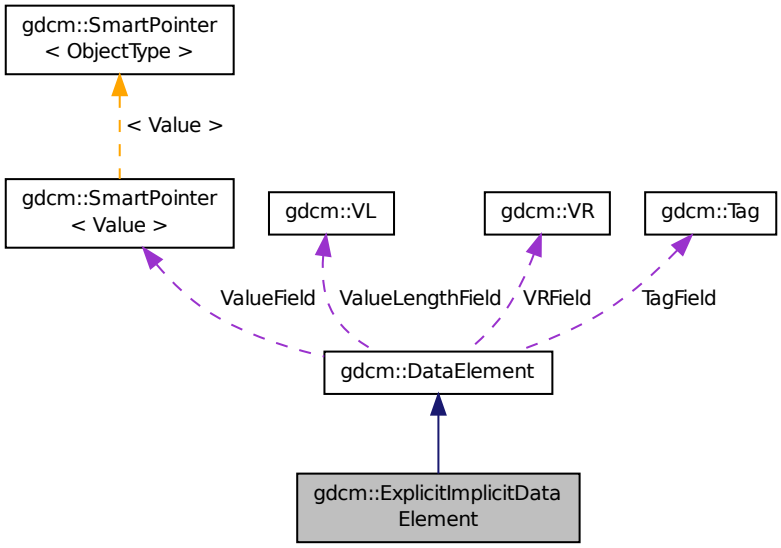
Class to read/write a DataElement as ExplicitImplicit Data Element.

```
#include <gdcmExplicitImplicitDataElement.h>
```

Inheritance diagram for gdcm::ExplicitImplicitDataElement:



Collaboration diagram for gdcm::ExplicitImplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)

- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)

Additional Inherited Members

25.108.1 Detailed Description

Class to read/write a DataElement as ExplicitImplicit Data Element.

Note

This only happen for some Philips images Should I derive from ExplicitDataElement instead ? This is the class that is the closest the GDCM1.x parser. At each element we try first to read it as explicit, if this fails, then we try again as an implicit element.

25.108.2 Member Function Documentation

25.108.2.1 VL gdcm::ExplicitImplicitDataElement::GetLength ( ) const

Reimplemented from gdcm::DataElement.

25.108.2.2 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::Read (std::istream & *is*)

Reimplemented from gdcm::DataElement.

25.108.2.3 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadPreValue (std::istream & *is*)

25.108.2.4 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadValue (std::istream & *is*)

25.108.2.5 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*) [inline]

Reimplemented from gdcm::DataElement.

The documentation for this class was generated from the following file:

- gdcmExplicitImplicitDataElement.h

25.109 gdcm::Fiducials Class Reference

Fiducials.

```
#include <gdcmFiducials.h>
```

Public Member Functions

- Fiducials ()

25.109.1 Detailed Description

Fiducials.

25.109.2 Constructor & Destructor Documentation

25.109.2.1 gdcm::Fiducials::Fiducials ( ) [inline]

The documentation for this class was generated from the following file:

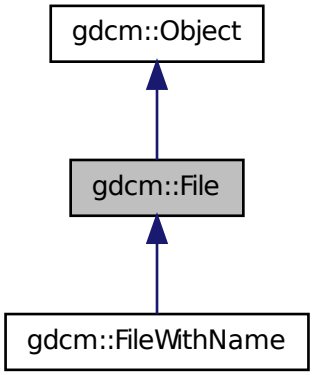
- gdcmFiducials.h

25.110 gdcm::File Class Reference

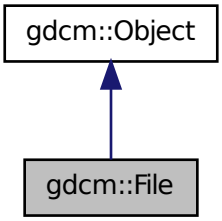
a DICOM File See PS 3.10 File: A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may by written, read and/or deleted.

`#include <gdcmFile.h>`

Inheritance diagram for gdcm::File:



Collaboration diagram for gdcm::File:



Public Member Functions

- File ()
- ~File ()
- const DataSet & GetDataSet () const

  Get Data Set.
- DataSet & GetDataSet ()

  Get Data Set.
- const FileMetaInformation & GetHeader () const

  Get File Meta Information.
- FileMetaInformation & GetHeader ()

  Get File Meta Information.
- std::istream & Read (std::istream &is)

  Read.
- void SetDataSet (const DataSet &ds)

  Set Data Set.
- void SetHeader (const FileMetaInformation &fmi)

  Set File Meta Information.
- std::ostream const & Write (std::ostream &os) const

  Write.

Friends

- std::ostream & operator<< (std::ostream &os, const File &val)

Additional Inherited Members

25.110.1 Detailed Description

a DICOM File See PS 3.10 File: A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may by written, read and/or deleted.

See also

Reader Writer

Examples:

ChangeSequenceUltrasound.cxx, CreateJPIPDataSet.cxx, DiffFile.cxx, DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, and StreamImageReaderTest.cxx.

25.110.2 Constructor & Destructor Documentation

25.110.2.1 gdcm::File::File ( ) [inline]

25.110.2.2 gdcm::File::~File ( ) [inline]

25.110.3 Member Function Documentation

25.110.3.1 const DataSet& gdcm::File::GetDataSet ( ) const [inline]

Get Data Set.

Examples:

ChangeSequenceUltrasound.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, EncapsulateFileInRawData.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

25.110.3.2 DataSet& gdcm::File::GetDataSet ( ) [inline]

Get Data Set.

25.110.3.3 const FileMetaInformation& gdcm::File::GetHeader ( ) const [inline]

Get File Meta Information.

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetJPEGSamplePrecision.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

Referenced by gdcm::operator<<().

25.110.3.4 FileMetaInformation& gdcm::File::GetHeader ( ) [inline]

Get File Meta Information.

25.110.3.5 std::istream& gdcm::File::Read (std::istream & *is*)

Read.

25.110.3.6 void gdcm::File::SetDataSet (const DataSet & *ds*) [inline]

Set Data Set.

25.110.3.7 void gdcm::File::SetHeader (const FileMetaInformation & *fmi*) [inline]

Set File Meta Information.

25.110.3.8 std::ostream const& gdcm::File::Write (std::ostream & *os*) const

Write.

25.110.4 Friends And Related Function Documentation

25.110.4.1 std::ostream& operator<< (std::ostream & *os*, const File & *val*) [friend]

The documentation for this class was generated from the following file:

- gdcmFile.h

25.111 gdcm::FileDerivation Class Reference

FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.

```
#include <gdcmFileDerivation.h>
```

Public Member Functions

- FileDerivation ()
- ~FileDerivation ()
- bool AddReference (const char *referencedsopclassuid, const char *referencedsopinstanceuid)
- bool Derive ()

  Change.
- File & GetFile ()
- const File & GetFile () const
- void SetDerivationCodeSequenceCodeValue (unsigned int codevalue)

  Specify the Derivation Code Sequence Code Value. Eg 113040.
- void SetDerivationDescription (const char *dd)

  Specify the Derivation Description. Eg "lossy conversion".

- void SetFile (const File &f)

  Set/Get File.
- void SetPurposeOfReferenceCodeSequenceCodeValue (unsigned int codevalue)

  Specify the Purpose Of Reference Code Value. Eg. 121320.

Protected Member Functions

- bool AddDerivationDescription ()
- bool AddPurposeOfReferenceCodeSequence (DataSet &ds)
- bool AddSourceImageSequence ()

25.111.1 Detailed Description

FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.

URL: `http://medical.nema.org/medical/dicom/2008/08_16pu.pdf`

DICOM Part 16 has two Context Groups CID 7202 and CID 7203 which contain a set of codes defining reason for a source image reference (ie. reason code for referenced image sequence) and a coded description of the deriation applied to the new image data from the original. Both these context groups are extensible.

File Derivation is compulsary when creating a lossy derived image.

Examples:

GenFakeImage.cxx.

25.111.2 Constructor & Destructor Documentation

25.111.2.1 gdcm::FileDerivation::FileDerivation ( )

25.111.2.2 gdcm::FileDerivation::∼FileDerivation ( )

25.111.3 Member Function Documentation

25.111.3.1 bool gdcm::FileDerivation::AddDerivationDescription ( ) [protected]

25.111.3.2 bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (DataSet & *ds*) [protected]

25.111.3.3 bool gdcm::FileDerivation::AddReference (const char ∗ *referencedsopclassuid,* const char ∗ *referencedsopinstanceuid*)

Create the proper reference. Need to pass the original SOP Class UID and the original SOP Instance UID, so that those value can be used as Reference.

Warning

referencedsopclassuid and referencedsopinstanceuid needs to be \0 padded. This is not compatible with how ByteValue->GetPointer works.

Examples:

GenFakeImage.cxx.

25.111.3.4 bool gdcm::FileDerivation::AddSourceImageSequence ( ) [protected]

25.111.3.5 bool gdcm::FileDerivation::Derive ( )

Change.

Examples:

GenFakeImage.cxx.

25.111.3.6 File& gdcm::FileDerivation::GetFile ( ) [inline]

Examples:

GenFakeImage.cxx.

25.111.3.7 const File& gdcm::FileDerivation::GetFile ( ) const [inline]

25.111.3.8 void gdcm::FileDerivation::SetDerivationCodeSequenceCodeValue (unsigned int *codevalue*)

Specify the Derivation Code Sequence Code Value. Eg 113040.

Examples:

GenFakeImage.cxx.

25.111.3.9 void gdcm::FileDerivation::SetDerivationDescription (const char * *dd*)

Specify the Derivation Description. Eg "lossy conversion".

25.111.3.10 void gdcm::FileDerivation::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

GenFakeImage.cxx.

25.111.3.11 void gdcm::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (unsigned int *codevalue*)

Specify the Purpose Of Reference Code Value. Eg. 121320.

Examples:

GenFakeImage.cxx.

The documentation for this class was generated from the following file:

- gdcmFileDerivation.h

25.112 gdcm::FileExplicitFilter Class Reference

FileExplicitFilter class After changing a file from Implicit to Explicit representation (see ImageChangeTransferSyntax) one operation is to make sure the VR of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit reprensentation, the VR is not stored directly in the file.

```
#include <gdcmFileExplicitFilter.h>
```

Public Member Functions

- FileExplicitFilter ()
- ~FileExplicitFilter ()
- bool Change ()

  Set FMI Transfer Syntax.
- File & GetFile ()
- void SetChangePrivateTags (bool b)

  Decide whether or not to VR'ify private tags.
- void SetFile (const File &f)

  Set/Get File.
- void SetRecomputeItemLength (bool b)

  By default set Sequence & Item length to Undefined to avoid recomputing length:
- void SetRecomputeSequenceLength (bool b)
- void SetUseVRUN (bool b)

  When VR=16bits in explicit but Implicit has a 32bits length, use VR=UN.

Protected Member Functions

- bool ChangeFMI ()
- bool ProcessDataSet (DataSet &ds, Dicts const &dicts)

25.112.1 Detailed Description

FileExplicitFilter class After changing a file from Implicit to Explicit representation (see ImageChangeTransferSyntax) one operation is to make sure the VR of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit reprensentation, the VR is not stored directly in the file.

Warning

changing an implicit dataset to an explicit dataset is NOT a trivial task of simply changing the VR to the dict one:

- One has to make sure SQ is properly set
- One has to recompute the explicit length SQ
- One has to make sure that VR is valid for the encoding
- One has to make sure that VR 16bits can store the original value length

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

25.112.2 Constructor & Destructor Documentation

25.112.2.1 gdcm::FileExplicitFilter::FileExplicitFilter ( ) [inline]

25.112.2.2 gdcm::FileExplicitFilter::~FileExplicitFilter ( ) [inline]

25.112.3 Member Function Documentation

25.112.3.1 bool gdcm::FileExplicitFilter::Change ( )

Set FMI Transfer Syntax.

Change

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

25.112.3.2 bool gdcm::FileExplicitFilter::ChangeFMI ( ) [protected]

25.112.3.3 File& gdcm::FileExplicitFilter::GetFile ( ) [inline]

25.112.3.4 bool gdcm::FileExplicitFilter::ProcessDataSet (DataSet & *ds*, Dicts const & *dicts*) [protected]

25.112.3.5 void gdcm::FileExplicitFilter::SetChangePrivateTags (bool *b*) [inline]

Decide whether or not to VR'ify private tags.

25.112.3.6 void gdcm::FileExplicitFilter::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

25.112.3.7 void gdcm::FileExplicitFilter::SetRecomputeItemLength (bool *b*)

By default set Sequence & Item length to Undefined to avoid recomputing length:

25.112.3.8 void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (bool *b*)

25.112.3.9 void gdcm::FileExplicitFilter::SetUseVRUN (bool *b*) [inline]

When VR=16bits in explicit but Implicit has a 32bits length, use VR=UN.

The documentation for this class was generated from the following file:

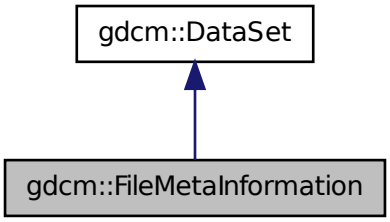
- gdcmFileExplicitFilter.h

25.113 gdcm::FileMetaInformation Class Reference

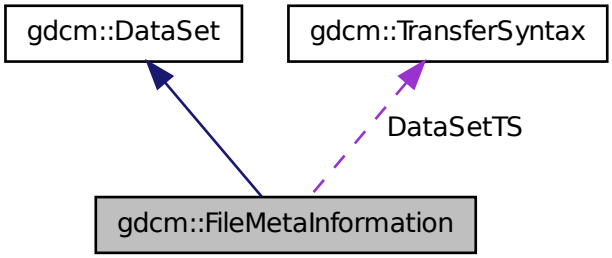
Class to represent a File Meta Information.

```
#include <gdcmFileMetaInformation.h>
```

Inheritance diagram for gdcm::FileMetaInformation:



Collaboration diagram for gdcm::FileMetaInformation:



Public Member Functions

- FileMetaInformation ()
- FileMetaInformation (FileMetaInformation const &fmi)
- ~FileMetaInformation ()
- void FillFromDataSet (DataSet const &ds)

  Construct a FileMetaInformation from an already existing DataSet:
- const TransferSyntax & GetDataSetTransferSyntax () const
- VL GetFullLength () const
- MediaStorage GetMediaStorage () const
- TransferSyntax::NegociatedType GetMetaInformationTS () const

- const Preamble & GetPreamble () const

  Get Preamble.
- Preamble & GetPreamble ()
- void Insert (const DataElement &de)
- bool IsValid () const
- std::istream & Read (std::istream &is)

  Read.
- std::istream & ReadCompat (std::istream &is)
- void Replace (const DataElement &de)

  Replace a dataelement with another one.
- void SetDataSetTransferSyntax (const TransferSyntax &ts)
- void SetPreamble (const Preamble &p)
- std::ostream & Write (std::ostream &os) const

  Write.

Static Public Member Functions

- static void AppendImplementationClassUID (const char ∗imp)
- static const char ∗ GetImplementationClassUID ()
- static const char ∗ GetImplementationVersionName ()
- static const char ∗ GetSourceApplicationEntityTitle ()
- static void SetImplementationClassUID (const char ∗imp)

  Override the GDCM default values:
- static void SetImplementationVersionName (const char ∗version)
- static void SetSourceApplicationEntityTitle (const char ∗title)

Protected Member Functions

- void ComputeDataSetMediaStorageSOPClass ()
- void ComputeDataSetTransferSyntax ()
- void Default ()
- template<typename TSwap >
  std::istream & ReadCompatInternal (std::istream &is)

Static Protected Member Functions

- static const char ∗ GetFileMetaInformationVersion ()
- static const char ∗ GetGDCMImplementationClassUID ()
- static const char ∗ GetGDCMImplementationVersionName ()
- static const char ∗ GetGDCMSourceApplicationEntityTitle ()

Protected Attributes

- MediaStorage::MSType DataSetMS
- TransferSyntax DataSetTS
- TransferSyntax::NegociatedType MetaInformationTS

Friends

- std::ostream & operator<< (std::ostream &_os, const FileMetaInformation &_val)

Additional Inherited Members

25.113.1 Detailed Description

Class to represent a File Meta Information.

FileMetaInformation is a Explicit Structured Set. Whenever the file contains an ImplicitDataElement DataSet, a conversion will take place.

Definition: The File Meta Information includes identifying information on the encapsulated Data Set. This header consists of a 128 byte File Preamble, followed by a 4 byte DICOM prefix, followed by the File Meta Elements shown in Table 7.1-1. This header shall be present in every DICOM file.

See also

Writer Reader

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

25.113.2 Constructor & Destructor Documentation

25.113.2.1 gdcm::FileMetaInformation::FileMetaInformation ( ) [inline]

25.113.2.2 gdcm::FileMetaInformation::~FileMetaInformation ( ) [inline]

25.113.2.3 gdcm::FileMetaInformation::FileMetaInformation (FileMetaInformation const & *fmi*) [inline]

References DataSetMS, DataSetTS, and MetaInformationTS.

25.113.3 Member Function Documentation

25.113.3.1 static void gdcm::FileMetaInformation::AppendImplementationClassUID (const char * *imp*) [static]

25.113.3.2 void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ( ) [protected]

25.113.3.3 void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ( ) [protected]

25.113.3.4 void gdcm::FileMetaInformation::Default ( ) [protected]

25.113.3.5 void gdcm::FileMetaInformation::FillFromDataSet (DataSet const & *ds*)

Construct a FileMetaInformation from an already existing DataSet:

25.113.3.6 const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax ( ) const [inline]

Examples:

GetJPEGSamplePrecision.cxx, and MergeTwoFiles.cxx.

25.113.3.7 static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion ( ) [static], [protected]

25.113.3.8 VL gdcm::FileMetaInformation::GetFullLength ( ) const [inline]

References gdcm::VL::GetLength().

25.113.3.9 static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID ( ) [static], [protected]

25.113.3.10 static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName ( ) [static], [protected]

25.113.3.11 static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ( ) [static], [protected]

25.113.3.12 static const char* gdcm::FileMetaInformation::GetImplementationClassUID ( ) [static]

25.113.3.13 static const char* gdcm::FileMetaInformation::GetImplementationVersionName ( ) [static]

25.113.3.14 MediaStorage gdcm::FileMetaInformation::GetMediaStorage ( ) const

25.113.3.15 TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS ( ) const [inline]

25.113.3.16 const Preamble& gdcm::FileMetaInformation::GetPreamble ( ) const [inline]

Get Preamble.

Referenced by gdcm::operator<<().

25.113.3.17 Preamble& gdcm::FileMetaInformation::GetPreamble ( ) [inline]

25.113.3.18 static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ( ) [static]

25.113.3.19 void gdcm::FileMetaInformation::Insert (const DataElement & *de*) [inline]

Insert a DataElement in the DataSet.

Warning

: Tag need to be >= 0x8 to be considered valid data element

Reimplemented from gdcm::DataSet.

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

25.113.3.20 bool gdcm::FileMetaInformation::IsValid ( ) const [inline]

25.113.3.21 std::istream& gdcm::FileMetaInformation::Read (std::istream & *is*)

Read.

Reimplemented from gdcm::DataSet.

25.113.3.22 std::istream& gdcm::FileMetaInformation::ReadCompat (std::istream & *is*)

25.113.3.23 template<typename TSwap > std::istream& gdcm::FileMetaInformation::ReadCompatInternal (std::istream & *is*) [protected]

25.113.3.24 void gdcm::FileMetaInformation::Replace (const **DataElement** & *de*) [inline]

Replace a dataelement with another one.

Reimplemented from gdcm::DataSet.

Examples:

LargeVRDSExplicit.cxx.

References gdcm::DataElement::GetTag().

25.113.3.25 void gdcm::FileMetaInformation::SetDataSetTransferSyntax (const **TransferSyntax** & *ts*)

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_-Stream_Image_Writer.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.-cxx, pmsct_rgb1.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

25.113.3.26 static void gdcm::FileMetaInformation::SetImplementationClassUID (const char * *imp*) [static]

Override the GDCM default values:

25.113.3.27 static void gdcm::FileMetaInformation::SetImplementationVersionName (const char * *version*) [static]

25.113.3.28 void gdcm::FileMetaInformation::SetPreamble (const **Preamble** & *p*) [inline]

25.113.3.29 static void gdcm::FileMetaInformation::SetSourceApplicationEntityTitle (const char * *title*) [static]

Examples:

FixJAIBugJPEGLS.cxx.

25.113.3.30 std::ostream& gdcm::FileMetaInformation::Write (std::ostream & *os*) const

Write.

Reimplemented from gdcm::DataSet.

25.113.4 Friends And Related Function Documentation

25.113.4.1 std::ostream& operator<< (std::ostream & _os, const FileMetaInformation & _val) [friend]

25.113.5 Member Data Documentation

25.113.5.1 MediaStorage::MSType gdcm::FileMetaInformation::DataSetMS [protected]

Referenced by FileMetaInformation().

25.113.5.2 TransferSyntax gdcm::FileMetaInformation::DataSetTS [protected]

Referenced by FileMetaInformation().

25.113.5.3 TransferSyntax::NegociatedType gdcm::FileMetaInformation::MetaInformationTS [protected]

Referenced by FileMetaInformation().

The documentation for this class was generated from the following file:

- gdcmFileMetaInformation.h

25.114 gdcm::Filename Class Reference

Class to manipulate file name's.

```
#include <gdcmFilename.h>
```

Public Member Functions

- Filename (const char *filename="")
- const char * GetExtension ()

  return only the extension part of a filename
- const char * GetFileName () const

  Return the full filename.
- const char * GetName ()

  return only the name part of a filename
- const char * GetPath ()

  Return only the path component of a filename.
- bool IsEmpty () const

  return whether the filename is empty
- bool IsIdentical (Filename const &fn) const
- operator const char * () const
- const char * ToUnixSlashes ()

  Convert backslash (windows style) to UNIX style slash.
- const char * ToWindowsSlashes ()

  Convert foward slash (UNIX style) to windows style slash.

Static Public Member Functions

- static const char ∗ Join (const char ∗path, const char ∗filename)

25.114.1 Detailed Description

Class to manipulate file name's.

Note

OS independant representation of a filename (to query path, name and extension from a filename)

25.114.2 Constructor & Destructor Documentation

25.114.2.1 gdcm::Filename::Filename (const char ∗ *filename* = " ") [inline]

25.114.3 Member Function Documentation

25.114.3.1 const char∗ gdcm::Filename::GetExtension ( )

return only the extension part of a filename

25.114.3.2 const char∗ gdcm::Filename::GetFileName ( ) const [inline]

Return the full filename.

25.114.3.3 const char∗ gdcm::Filename::GetName ( )

return only the name part of a filename

25.114.3.4 const char∗ gdcm::Filename::GetPath ( )

Return only the path component of a filename.

25.114.3.5 bool gdcm::Filename::IsEmpty ( ) const [inline]

return whether the filename is empty

25.114.3.6 bool gdcm::Filename::IsIdentical (Filename const & *fn*) const

25.114.3.7 static const char∗ gdcm::Filename::Join (const char ∗ *path,* const char ∗ *filename*) [static]

Join two paths NOT THREAD SAFE

25.114.3.8 gdcm::Filename::operator const char ∗ ( ) const [inline]

Simple operator to allow Filename myfilename("..."); const char ∗ s = myfilename;

25.114.3.9 const char∗ gdcm::Filename::ToUnixSlashes ( )

Convert backslash (windows style) to UNIX style slash.

25.114.3.10 const char∗ gdcm::Filename::ToWindowsSlashes ( )

Convert foward slash (UNIX style) to windows style slash.

The documentation for this class was generated from the following file:

- gdcmFilename.h

25.115 gdcm::FilenameGenerator Class Reference

FilenameGenerator.

```
#include <gdcmFilenameGenerator.h>
```

Public Types

- typedef std::vector< FilenameType > FilenamesType
- typedef std::string FilenameType
- typedef FilenamesType::size_type SizeType

Public Member Functions

- FilenameGenerator ()
- ∼FilenameGenerator ()
- bool Generate ()

  Generate (return success)
- const char ∗ GetFilename (SizeType n) const

  Get a particular filename (call after Generate)
- FilenamesType const & GetFilenames () const

  Return all filenames.
- SizeType GetNumberOfFilenames () const
- const char ∗ GetPattern () const
- const char ∗ GetPrefix () const
- void SetNumberOfFilenames (SizeType nfiles)

  Set/Get the number of filenames to generate.
- void SetPattern (const char ∗pattern)

  Set/Get pattern.
- void SetPrefix (const char ∗prefix)

  Set/Get prefix.

25.115.1 Detailed Description

FilenameGenerator.

class to generate filenames based on a pattern (C-style)

Output will be:

for i = 0, number of filenames: outfilename[i] = prefix + (pattern % i)

where pattern % i means C-style snprintf of Pattern using value 'i'

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.115.2 Member Typedef Documentation

25.115.2.1 typedef std::vector<FilenameType> gdcm::FilenameGenerator::FilenamesType

25.115.2.2 typedef std::string gdcm::FilenameGenerator::FilenameType

25.115.2.3 typedef FilenamesType::size_type gdcm::FilenameGenerator::SizeType

25.115.3 Constructor & Destructor Documentation

25.115.3.1 gdcm::FilenameGenerator::FilenameGenerator ( ) [inline]

25.115.3.2 gdcm::FilenameGenerator::~FilenameGenerator ( ) [inline]

25.115.4 Member Function Documentation

25.115.4.1 bool gdcm::FilenameGenerator::Generate ( )

Generate (return success)

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.115.4.2 const char* gdcm::FilenameGenerator::GetFilename (SizeType *n*) const

Get a particular filename (call after Generate)

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.115.4.3 FilenamesType const& gdcm::FilenameGenerator::GetFilenames ( ) const [inline]

Return all filenames.

25.115.4.4 SizeType gdcm::FilenameGenerator::GetNumberOfFilenames ( ) const

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.115.4.5 const char* gdcm::FilenameGenerator::GetPattern ( ) const [inline]

25.115.4.6 const char* gdcm::FilenameGenerator::GetPrefix ( ) const [inline]

25.115.4.7 void gdcm::FilenameGenerator::SetNumberOfFilenames (SizeType *nfiles*)

Set/Get the number of filenames to generate.

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.115.4.8 void gdcm::FilenameGenerator::SetPattern (const char * *pattern*) [inline]

Set/Get pattern.

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.115.4.9 void gdcm::FilenameGenerator::SetPrefix (const char * *prefix*) [inline]

Set/Get prefix.

The documentation for this class was generated from the following file:

- gdcmFilenameGenerator.h

25.116 gdcm::FileSet Class Reference

File-set: A File-set is a collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which File IDs are unique.

```
#include <gdcmFileSet.h>
```

Public Types

- typedef std::vector< FileType > FilesType
- typedef std::string FileType

Public Member Functions

- FileSet ()
- void AddFile (File const &)
- bool AddFile (const char *filename)
- FilesType const & GetFiles () const
- void SetFiles (FilesType const &files)

Friends

- std::ostream & operator<< (std::ostream &_os, const FileSet &d)

25.116.1 Detailed Description

File-set: A File-set is a collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which File IDs are unique.

25.116.2 Member Typedef Documentation

25.116.2.1 typedef std::vector<FileType> gdcm::FileSet::FilesType

25.116.2.2 typedef std::string gdcm::FileSet::FileType

25.116.3 Constructor & Destructor Documentation

25.116.3.1 gdcm::FileSet::FileSet ( ) [inline]

25.116.4 Member Function Documentation

25.116.4.1 void gdcm::FileSet::AddFile (File const &) [inline]

Deprecated . Does nothing

25.116.4.2 bool gdcm::FileSet::AddFile (const char * *filename*)

Add a file 'filename' to the list of files. Return true on success, false in case filename could not be found on system.

25.116.4.3 FilesType const& gdcm::FileSet::GetFiles ( ) const [inline]

25.116.4.4 void gdcm::FileSet::SetFiles (FilesType const & *files*)

25.116.5 Friends And Related Function Documentation

25.116.5.1 std::ostream& operator<< (std::ostream & *_os,* const FileSet & *d*) [friend]

The documentation for this class was generated from the following file:

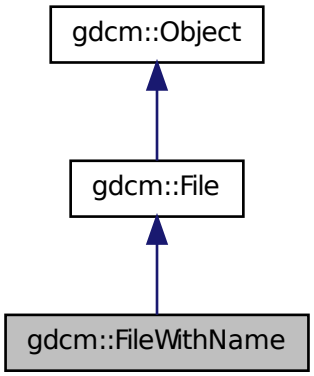
- gdcmFileSet.h

25.117 gdcm::FileWithName Class Reference

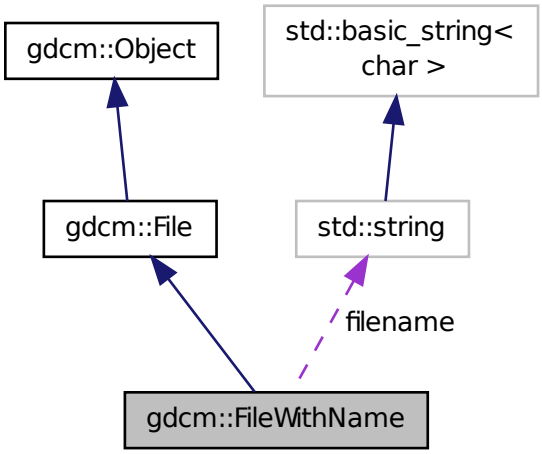
FileWithName.

```
#include <gdcmSerieHelper.h>
```

Inheritance diagram for gdcm::FileWithName:



Collaboration diagram for gdcm::FileWithName:



Public Member Functions

- FileWithName (File &f)

Public Attributes

- std::string filename

25.117.1 Detailed Description

FileWithName.

Backward only class do not use in newer code

25.117.2 Constructor & Destructor Documentation

25.117.2.1 gdcm::FileWithName::FileWithName (File & *f*) [inline]

25.117.3 Member Data Documentation

25.117.3.1 std::string gdcm::FileWithName::filename

The documentation for this class was generated from the following file:

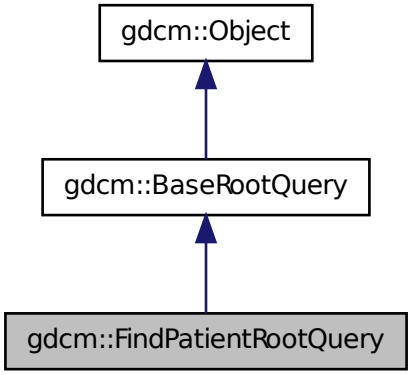
- gdcmSerieHelper.h

25.118 gdcm::FindPatientRootQuery Class Reference

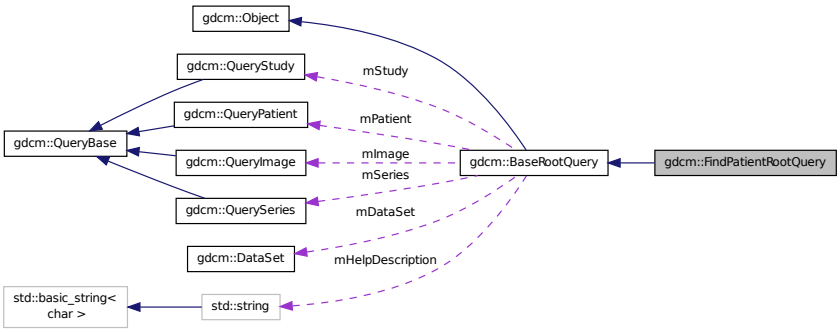
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

```
#include <gdcmFindPatientRootQuery.h>
```

Inheritance diagram for gdcm::FindPatientRootQuery:



Collaboration diagram for gdcm::FindPatientRootQuery:



Public Member Functions

- FindPatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

25.118.1 Detailed Description

PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

25.118.2 Constructor & Destructor Documentation

25.118.2.1 gdcm::FindPatientRootQuery::FindPatientRootQuery ( )

25.118.3 Member Function Documentation

25.118.3.1 UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseRootQuery.

25.118.3.2 std::vector<Tag> gdcm::FindPatientRootQuery::GetTagListByLevel (const EQueryLevel & *inQueryLevel*) [virtual]

this function will return all tags at a given query level, so that ∗they maybe selected for searching. The boolean forFind is true ∗if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

25.118.3.3 void gdcm::FindPatientRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [virtual]

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

25.118.3.4 bool gdcm::FindPatientRootQuery::ValidateQuery (bool *inStrict* = true) const [virtual]

have to be able to ensure that ∗0x8,0x52 is set (which will be true if InitializeDataSet is called...) ∗that the level is appropriate (ie, not setting PATIENT for a study query ∗that the tags in the query match the right level (either required, unique, optional) ∗by default, this function checks to see if the query is for finding, which is more ∗permissive than for moving. For moving, only the unique tags are allowed. ∗10 Jan 2011: adding in the 'strict' mode. ∗according to the standard (at least, how I've read it), only tags for a particular ∗level should be allowed in a particular query (ie, just series level tags in a series ∗level query). However, it seems that dcm4chee doesn't share that interpretation. ∗So, if 'inStrict' is false, then tags from the current level and all higher levels ∗are now considered valid. So, if you're doing a non-strict series-level query, ∗tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

25.118.4 Friends And Related Function Documentation

25.118.4.1 friend class QueryFactory [friend]

Reimplemented from gdcm::BaseRootQuery.

The documentation for this class was generated from the following file:

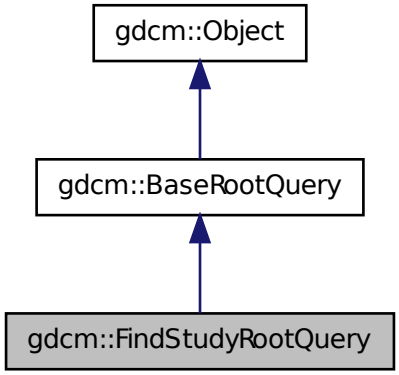
- gdcmFindPatientRootQuery.h

25.119 gdcm::FindStudyRootQuery Class Reference

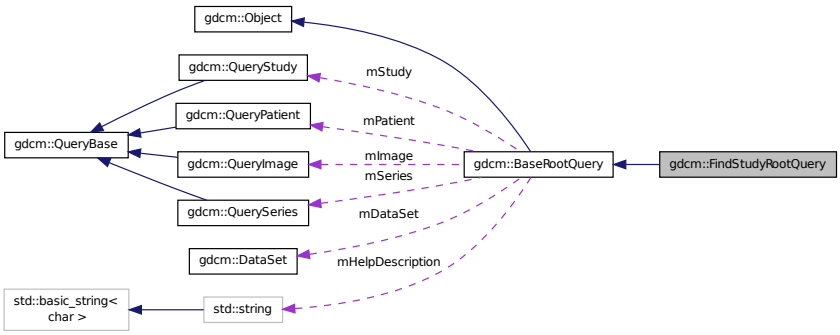
FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

```
#include <gdcmFindStudyRootQuery.h>
```

Inheritance diagram for gdcm::FindStudyRootQuery:



Collaboration diagram for gdcm::FindStudyRootQuery:



Public Member Functions

- FindStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)

- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

25.119.1 Detailed Description

FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

25.119.2 Constructor & Destructor Documentation

25.119.2.1 gdcm::FindStudyRootQuery::FindStudyRootQuery ( )

25.119.3 Member Function Documentation

25.119.3.1 UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseRootQuery.

25.119.3.2 std::vector<Tag> gdcm::FindStudyRootQuery::GetTagListByLevel (const EQueryLevel & *inQueryLevel*) [virtual]

this function will return all tags at a given query level, so that ∗they maybe selected for searching. The boolean forFind is true ∗if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

25.119.3.3 void gdcm::FindStudyRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [virtual]

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

25.119.3.4 bool gdcm::FindStudyRootQuery::ValidateQuery (bool *inStrict* = true) const [virtual]

have to be able to ensure that (0008,0052) is set that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional)

Implements gdcm::BaseRootQuery.

25.119.4 Friends And Related Function Documentation

25.119.4.1 friend class QueryFactory [friend]

Reimplemented from gdcm::BaseRootQuery.

The documentation for this class was generated from the following file:

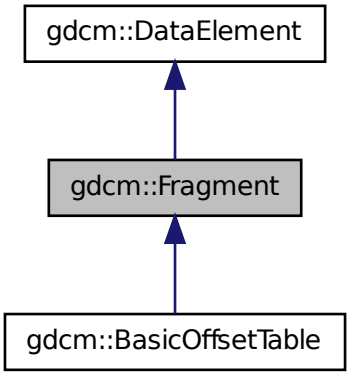
- gdcmFindStudyRootQuery.h

25.120 gdcm::Fragment Class Reference

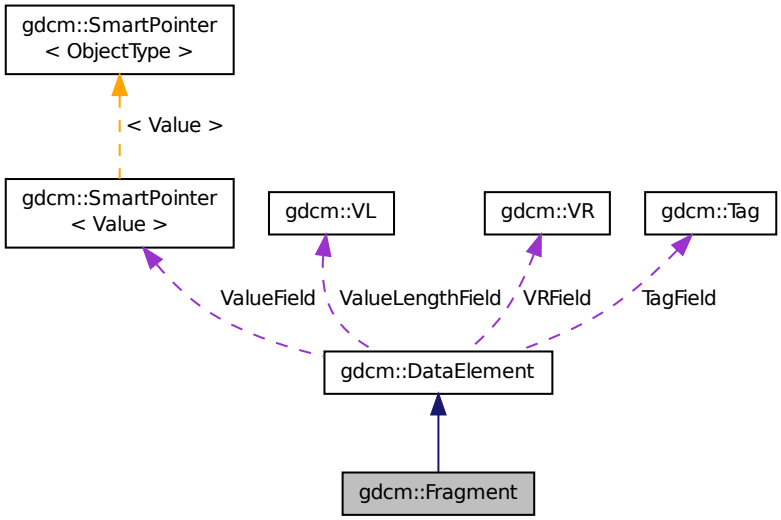
Class to represent a Fragment.

`#include <gdcmFragment.h>`

Inheritance diagram for gdcm::Fragment:



Collaboration diagram for gdcm::Fragment:



Public Member Functions

- Fragment ()
- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)
- template<typename TSwap >
  std::ostream & Write (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &os, const Fragment &val)

Additional Inherited Members

25.120.1 Detailed Description

Class to represent a Fragment.

Examples:

FixBrokenJ2K.cxx, and FixJAIBugJPEGLS.cxx.

25.120.2 Constructor & Destructor Documentation

25.120.2.1 gdcm::Fragment::Fragment ( ) [inline]

25.120.3 Member Function Documentation

25.120.3.1 VL gdcm::Fragment::GetLength ( ) const [inline]

Reimplemented from gdcm::DataElement.

References gdcm::VL::GetLength().

25.120.3.2 template<typename TSwap > std::istream& gdcm::Fragment::Read (std::istream & *is*) [inline]

Reimplemented from gdcm::DataElement.

Reimplemented in gdcm::BasicOffsetTable.

References gdcm::VL::Read().

Referenced by gdcm::SequenceOfFragments::Read().

25.120.3.3 template<typename TSwap > std::istream& gdcm::Fragment::ReadValue (std::istream & *is*) [inline]

References gdcmWarningMacro, and gdcm::ParseException::SetLastElement().

25.120.3.4 template<typename TSwap > std::ostream& gdcm::Fragment::Write (std::ostream & *os*) const [inline]

Reimplemented from gdcm::DataElement.

References gdcm::ByteValue::GetLength(), gdcm::VL::Write(), and gdcm::ByteValue::Write().

25.120.4 Friends And Related Function Documentation

25.120.4.1 std::ostream& operator<< (std::ostream & *os*, const Fragment & *val*) [friend]

The documentation for this class was generated from the following file:

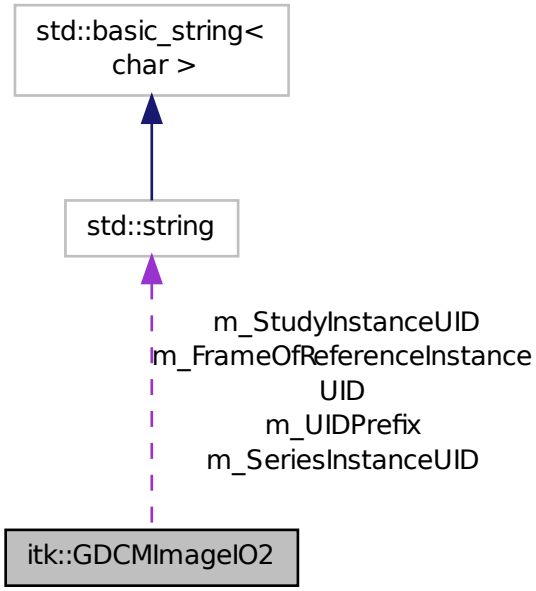
- gdcmFragment.h

25.121 itk::GDCMImageIO2 Class Reference

ImageIO class for reading and writing DICOM V3.0 and ACR/NEMA (V1.0 & V2.0) images This class is only an adaptor to the gdcm library (currently gdcm 2.0 is used):

```
#include <itkGDCMImageIO2.h>
```

Collaboration diagram for itk::GDCMImageIO2:



Public Types

- typedef SmartPointer< Self > Pointer
- typedef GDCMImageIO2 Self
- typedef ImageIOBase Superclass
- enum TCompressionType {
  JPEG = 0,
  JPEG2000 }

Public Member Functions

- virtual bool CanReadFile (const char *)
- virtual bool CanWriteFile (const char *)
- void GetBodyPart (char *part)
- void GetInstitution (char *ins)
- void GetManufacturer (char *manu)
- void GetModality (char *modality)
- void GetModel (char *model)
- void GetNumberOfSeriesInStudy (char *series)
- void GetNumberOfStudyRelatedSeries (char *series)
- void GetPatientAge (char *age)
- void GetPatientDOB (char *dob)
- void GetPatientID (char *id)

- void GetPatientName (char ∗name)
- void GetPatientSex (char ∗sex)
- void GetScanOptions (char ∗options)
- void GetStudyDate (char ∗date)
- void GetStudyDescription (char ∗desc)
- void GetStudyID (char ∗id)
- bool GetValueFromTag (const std::string &tag, std::string &value)
- itkBooleanMacro (KeepOriginalUID)
- itkBooleanMacro (LoadSequences)
- itkBooleanMacro (LoadPrivateTags)
- itkGetEnumMacro (CompressionType, TCompressionType)
- itkGetMacro (RescaleSlope, double)
- itkGetMacro (RescaleIntercept, double)
- itkGetMacro (KeepOriginalUID, bool)
- itkGetMacro (LoadSequences, bool)
- itkGetMacro (LoadPrivateTags, bool)
- itkGetStringMacro (UIDPrefix)
- itkGetStringMacro (StudyInstanceUID)
- itkGetStringMacro (SeriesInstanceUID)
- itkGetStringMacro (FrameOfReferenceInstanceUID)
- itkNewMacro (Self)
- itkSetEnumMacro (CompressionType, TCompressionType)
- itkSetMacro (KeepOriginalUID, bool)
- itkSetMacro (MaxSizeLoadEntry, long)
- itkSetMacro (LoadSequences, bool)
- itkSetMacro (LoadPrivateTags, bool)
- itkSetStringMacro (UIDPrefix)
- itkTypeMacro (GDCMImageIO2, Superclass)
- virtual void Read (void ∗buffer)
- virtual void ReadImageInformation ()
- virtual void Write (const void ∗buffer)
- virtual void WriteImageInformation ()

Static Public Member Functions

- static bool GetLabelFromTag (const std::string &tag, std::string &labelId)
- static bool GetLoadPrivateTagsDefault ()
- static bool GetLoadSequencesDefault ()
- static void LoadPrivateTagsDefaultOff ()
- static void LoadPrivateTagsDefaultOn ()
- static void LoadSequencesDefaultOff ()
- static void LoadSequencesDefaultOn ()
- static void SetLoadPrivateTagsDefault (bool)
- static void SetLoadSequencesDefault (bool)

Protected Member Functions

- GDCMImageIO2 ()
- ~GDCMImageIO2 ()
- void InternalReadImageInformation (std::ifstream &)
- bool OpenGDCMFileForReading (std::ifstream &, const char *)
- bool OpenGDCMFileForWriting (std::ofstream &, const char *)
- void PrintSelf (std::ostream &os, Indent indent) const

Protected Attributes

- std::string m_FrameOfReferenceInstanceUID
- bool m_KeepOriginalUID
- double m_RescaleIntercept
- double m_RescaleSlope
- std::string m_SeriesInstanceUID
- std::string m_StudyInstanceUID
- std::string m_UIDPrefix

25.121.1 Detailed Description

ImageIO class for reading and writing DICOM V3.0 and ACR/NEMA (V1.0 & V2.0) images This class is only an adaptor to the gdcm library (currently gdcm 2.0 is used):

`http://gdcm.sourceforge.net`

Warning

this class is deprecated, as gdcm 2.x has been integrated in ITK starting ITK 3.12

25.121.2 Member Typedef Documentation

25.121.2.1 typedef SmartPointer<Self> itk::GDCMImageIO2::Pointer

25.121.2.2 typedef GDCMImageIO2 itk::GDCMImageIO2::Self

Standard class typedefs.

25.121.2.3 typedef ImageIOBase itk::GDCMImageIO2::Superclass

25.121.3 Member Enumeration Documentation

25.121.3.1 enum itk::GDCMImageIO2::TCompressionType

Set/Get a boolean to use the JPEG2000 compression or not.

Enumerator:

JPEG

JPEG2000

25.121.4 Constructor & Destructor Documentation

25.121.4.1 itk::GDCMImageIO2::GDCMImageIO2 ( ) [protected]

25.121.4.2 itk::GDCMImageIO2::~GDCMImageIO2 ( ) [protected]

25.121.5 Member Function Documentation

25.121.5.1 virtual bool itk::GDCMImageIO2::CanReadFile (const char *) [virtual]

Determine the file type. Returns true if this ImageIO can read the file specified.

25.121.5.2 virtual bool itk::GDCMImageIO2::CanWriteFile (const char *) [virtual]

Determine the file type. Returns true if this ImageIO can write the file specified. GDCM triggers on ".dcm" and ".dicom".

25.121.5.3 void itk::GDCMImageIO2::GetBodyPart (char * *part*)

25.121.5.4 void itk::GDCMImageIO2::GetInstitution (char * *ins*)

25.121.5.5 static bool itk::GDCMImageIO2::GetLabelFromTag (const std::string & *tag,* std::string & *labelId*) [static]

Method for consulting the DICOM dictionary and recovering the text description of a field using its numeric tag represented as a string. If the tagkey is not found in the dictionary then this static method return false and the value "Unknown " in the labelId. If the tagkey is found then this static method returns true and the actual string descriptor of the tagkey is returned in the variable labelId.

25.121.5.6 static bool itk::GDCMImageIO2::GetLoadPrivateTagsDefault ( ) [inline],[static]

25.121.5.7 static bool itk::GDCMImageIO2::GetLoadSequencesDefault ( ) [inline],[static]

25.121.5.8 void itk::GDCMImageIO2::GetManufacturer (char * *manu*)

25.121.5.9 void itk::GDCMImageIO2::GetModality (char * *modality*)

25.121.5.10 void itk::GDCMImageIO2::GetModel (char * *model*)

25.121.5.11 void itk::GDCMImageIO2::GetNumberOfSeriesInStudy (char * *series*)

25.121.5.12 void itk::GDCMImageIO2::GetNumberOfStudyRelatedSeries (char * *series*)

25.121.5.13 void itk::GDCMImageIO2::GetPatientAge (char * *age*)

25.121.5.14 void itk::GDCMImageIO2::GetPatientDOB (char * *dob*)

25.121.5.15 void itk::GDCMImageIO2::GetPatientID (char * *id*)

25.121.5.16 void itk::GDCMImageIO2::GetPatientName (char * *name*)

Convenience methods to query patient information and scanner information. These methods are here for compatibility with the DICOMImageIO2 class.

25.121.5.17 void itk::GDCMImageIO2::GetPatientSex (char * *sex*)

25.121.5.18 void itk::GDCMImageIO2::GetScanOptions (char * *options*)

25.121.5.19 void itk::GDCMImageIO2::GetStudyDate (char * *date*)

25.121.5.20 void itk::GDCMImageIO2::GetStudyDescription (char * *desc*)

25.121.5.21 void itk::GDCMImageIO2::GetStudyID (char * *id*)

25.121.5.22 bool itk::GDCMImageIO2::GetValueFromTag (const std::string & *tag,* std::string & *value*)

More general method to retrieve an arbitrary DICOM value based on a DICOM Tag (eg "0123|4567"). WARNING: You need to use the lower case for hex 0x[a-f], for instance: "0020|000d" instead of "0020|000D" (the latter won't work)

25.121.5.23 void itk::GDCMImageIO2::InternalReadImageInformation (std::ifstream &) [protected]

25.121.5.24 itk::GDCMImageIO2::itkBooleanMacro (KeepOriginalUID)

25.121.5.25 itk::GDCMImageIO2::itkBooleanMacro (LoadSequences)

25.121.5.26 itk::GDCMImageIO2::itkBooleanMacro (LoadPrivateTags)

25.121.5.27 itk::GDCMImageIO2::itkGetEnumMacro (CompressionType , TCompressionType)

25.121.5.28 itk::GDCMImageIO2::itkGetMacro (RescaleSlope , double)

Macro to access Rescale Slope and Rescale Intercept. Which are needed to rescale properly image when needed. User then need to Always check those value when access value from the DICOM header

25.121.5.29 itk::GDCMImageIO2::itkGetMacro (RescaleIntercept , double)

25.121.5.30 itk::GDCMImageIO2::itkGetMacro (KeepOriginalUID , bool)

25.121.5.31 itk::GDCMImageIO2::itkGetMacro (LoadSequences , bool)

25.121.5.32 itk::GDCMImageIO2::itkGetMacro (LoadPrivateTags , bool)

25.121.5.33 itk::GDCMImageIO2::itkGetStringMacro (UIDPrefix)

Macro to access the DICOM UID prefix. By default this is the ITK root id. This default can be overriden if the exam is for example part of an existing study.

25.121.5.34 itk::GDCMImageIO2::itkGetStringMacro (StudyInstanceUID)

Access the generated DICOM UID's.

25.121.5.35 itk::GDCMImageIO2::itkGetStringMacro (SeriesInstanceUID)

25.121.5.36 itk::GDCMImageIO2::itkGetStringMacro (FrameOfReferenceInstanceUID)

25.121.5.37 itk::GDCMImageIO2::itkNewMacro (Self)

Method for creation through the object factory.

25.121.5.38 itk::GDCMImageIO2::itkSetEnumMacro (CompressionType , TCompressionType)

25.121.5.39 itk::GDCMImageIO2::itkSetMacro (KeepOriginalUID , bool)

Preserve the original DICOM UID of the input files

25.121.5.40 itk::GDCMImageIO2::itkSetMacro (MaxSizeLoadEntry , long)

A DICOM file can contains multiple binary stream that can be very long For example an Overlay on the image. Most of the time user do not want to load this binary structure in memory since it can consume lot of memory. Therefore any field that is bigger than the default value 0xfff is discarded and just seek'd This method allow advanced user to force the reading of such field

25.121.5.41 itk::GDCMImageIO2::itkSetMacro (LoadSequences , bool)

Parse any sequences in the DICOM file. Defaults to the value of LoadSequencesDefault. Loading DICOM files is faster when sequences are not needed.

25.121.5.42 itk::GDCMImageIO2::itkSetMacro (LoadPrivateTags , bool)

Parse any private tags in the DICOM file. Defaults to the value of LoadPrivateTagsDefault. Loading DICOM files is faster when private tags are not needed.

25.121.5.43 itk::GDCMImageIO2::itkSetStringMacro (UIDPrefix)

25.121.5.44 itk::GDCMImageIO2::itkTypeMacro (GDCMImageIO2 , Superclass)

Run-time type information (and related methods).

25.121.5.45 static void itk::GDCMImageIO2::LoadPrivateTagsDefaultOff ( ) [inline],[static]

25.121.5.46 static void itk::GDCMImageIO2::LoadPrivateTagsDefaultOn ( ) [inline],[static]

25.121.5.47 static void itk::GDCMImageIO2::LoadSequencesDefaultOff ( ) [inline],[static]

25.121.5.48 static void itk::GDCMImageIO2::LoadSequencesDefaultOn ( ) [inline],[static]

25.121.5.49 bool itk::GDCMImageIO2::OpenGDCMFileForReading (std::ifstream & , const char *) [protected]

25.121.5.50 bool itk::GDCMImageIO2::OpenGDCMFileForWriting (std::ofstream & , const char *) [protected]

25.121.5.51 void itk::GDCMImageIO2::PrintSelf (std::ostream & *os,* Indent *indent*) const [protected]

25.121.5.52 virtual void itk::GDCMImageIO2::Read (void * *buffer*) [virtual]

Reads the data from disk into the memory buffer provided.

25.121.5.53 virtual void itk::GDCMImageIO2::ReadImageInformation ( ) [virtual]

Set the spacing and dimesion information for the current filename.

25.121.5.54 static void itk::GDCMImageIO2::SetLoadPrivateTagsDefault (bool) [inline],[static]

Global method to define the default value for LoadPrivateTags. When instances of GDCMImageIO are created, the ivar LoadPrivateTags is initialized to the value of LoadPrivateTagsDefault. This method is useful when relying on the IO factory mechanism to load images rather than specifying a particular ImageIO object on the readers. Default is false.

25.121.5.55 static void itk::GDCMImageIO2::SetLoadSequencesDefault (bool) [inline],[static]

Global method to define the default value for LoadSequences. When instances of GDCMImageIO are created, the ivar LoadSequences is initialized to the value of LoadSequencesDefault. This method is useful when relying on the IO factory mechanism to load images rather than specifying a particular ImageIO object on the readers. Default is false.

25.121.5.56 virtual void itk::GDCMImageIO2::Write (const void * *buffer*) [virtual]

Writes the data to disk from the memory buffer provided. Make sure that the IORegion has been set properly.

25.121.5.57 virtual void itk::GDCMImageIO2::WriteImageInformation ( ) [virtual]

Writes the spacing and dimentions of the image. Assumes SetFileName has been called with a valid file name.

25.121.6 Member Data Documentation

25.121.6.1 std::string itk::GDCMImageIO2::m_FrameOfReferenceInstanceUID [protected]

25.121.6.2 bool itk::GDCMImageIO2::m_KeepOriginalUID [protected]

25.121.6.3 double itk::GDCMImageIO2::m_RescaleIntercept [protected]

25.121.6.4 double itk::GDCMImageIO2::m_RescaleSlope [protected]

25.121.6.5 std::string itk::GDCMImageIO2::m_SeriesInstanceUID [protected]

25.121.6.6 std::string itk::GDCMImageIO2::m_StudyInstanceUID [protected]

25.121.6.7 std::string itk::GDCMImageIO2::m_UIDPrefix [protected]

The documentation for this class was generated from the following file:

- itkGDCMImageIO2.h

25.122 gdcm::Global Class Reference

Global.

`#include <gdcmGlobal.h>`

Public Member Functions

- Global ()
- ~Global ()
- bool Append (const char *path)
- Defs const & GetDefs () const
- Dicts const & GetDicts () const
- Dicts & GetDicts ()
- bool LoadResourcesFiles ()
- bool Prepend (const char *path)

Static Public Member Functions

- static Global & GetInstance ()

  return the singleton instance

Protected Member Functions

- const char * Locate (const char *resfile) const

  Locate a ressource file.

Friends

- std::ostream & operator<< (std::ostream &_os, const Global &g)

25.122.1 Detailed Description

Global.

Note

Global should be included in any translation unit that will use Dict or that implements the singleton pattern. It makes sure that the Dict singleton is created before and destroyed after all other singletons in GDCM.

Examples:

GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

25.122.2 Constructor & Destructor Documentation

25.122.2.1 gdcm::Global::Global ( )

25.122.2.2 gdcm::Global::~Global ( )

25.122.3 Member Function Documentation

25.122.3.1 bool gdcm::Global::Append (const char * *path*)

Append path at the end of the path list

Warning

not thread safe !

25.122.3.2 Defs const& gdcm::Global::GetDefs ( ) const

retrieve the default/internal (Part 3) You need to explicitely call LoadResourcesFiles before

Examples:

GenerateStandardSOPClasses.cxx.

25.122.3.3 Dicts const& gdcm::Global::GetDicts ( ) const

retrieve the default/internal dicts (Part 6) This dict is filled up at load time

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, MrProtocol.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

25.122.3.4 Dicts& gdcm::Global::GetDicts ( )

25.122.3.5 static Global& gdcm::Global::GetInstance ( ) [static]

return the singleton instance

Examples:

GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, MrProtocol.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

25.122.3.6 bool gdcm::Global::LoadResourcesFiles ( )

Load all internal XML files, ressource path need to have been set before calling this member function (see Append/-Prepend members func)

Warning

not thread safe !

Examples:

GenerateStandardSOPClasses.cxx.

25.122.3.7 const char* gdcm::Global::Locate (const char * *resfile*) const [protected]

Locate a ressource file.

25.122.3.8 bool gdcm::Global::Prepend (const char * *path*)

Prepend path at the begining of the path list

Warning

not thread safe !

25.122.4 Friends And Related Function Documentation

25.122.4.1 std::ostream& operator<< (std::ostream & *_os*, const Global & *g*) [friend]

The documentation for this class was generated from the following file:

- gdcmGlobal.h

25.123 gdcm::GroupDict Class Reference

Class to represent the mapping from group number to its abbreviation and name.

```
#include <gdcmGroupDict.h>
```

Public Types

- typedef std::vector< std::string > GroupStringVector

Public Member Functions

- GroupDict ()
- ~GroupDict ()
- std::string const & GetAbbreviation (uint16_t num) const
- std::string const & GetName (uint16_t num) const
- size_t Size () const

Protected Member Functions

- void Add (std::string const &abbreviation, std::string const &name)
- void Insert (uint16_t num, std::string const &abbreviation, std::string const &name)

Friends

- std::ostream & operator<< (std::ostream &_os, const GroupDict &_val)

25.123.1 Detailed Description

Class to represent the mapping from group number to its abbreviation and name.

Note

Should I rewrite this class to use a std::map instead of std::vector for problem of memory consumption ?

25.123.2 Member Typedef Documentation

25.123.2.1 typedef std::vector<std::string> gdcm::GroupDict::GroupStringVector

25.123.3 Constructor & Destructor Documentation

25.123.3.1 gdcm::GroupDict::GroupDict ( ) [inline]

25.123.3.2 gdcm::GroupDict::~GroupDict ( ) [inline]

25.123.4 Member Function Documentation

25.123.4.1 void gdcm::GroupDict::Add (std::string const & *abbreviation*, std::string const & *name*) [protected]

25.123.4.2 std::string const& gdcm::GroupDict::GetAbbreviation (uint16_t *num*) const

Referenced by gdcm::operator<<().

25.123.4.3 std::string const& gdcm::GroupDict::GetName (uint16_t *num*) const

Referenced by gdcm::operator<<().

25.123.4.4 void gdcm::GroupDict::Insert (uint16_t *num*, std::string const & *abbreviation*, std::string const & *name*) [protected]

25.123.4.5 size_t gdcm::GroupDict::Size ( ) const [inline]

Referenced by gdcm::operator<<().

25.123.5 Friends And Related Function Documentation

25.123.5.1 std::ostream& operator<< (std::ostream & _os, const GroupDict & _val) [friend]

The documentation for this class was generated from the following file:

- gdcmGroupDict.h

25.124 gdcm::IconImageFilter Class Reference

IconImageFilter This filter will extract icons from a gdcm::File This filter will loop over all known sequence (public and private) that may contains an IconImage and retrieve them. The filter will fails with a value of false if no icon can be found Since it handle both public and private icon type, one should not assume the icon is in uncompress form, some private vendor store private icon in JPEG8/JPEG12.

```
#include <gdcmIconImageFilter.h>
```

Public Member Functions

- IconImageFilter ()
- ~IconImageFilter ()
- bool Extract ()

  Extract all Icon found in File.
- File & GetFile ()
- const File & GetFile () const
- IconImage & GetIconImage (unsigned int i) const
- unsigned int GetNumberOfIconImages () const

  Retrieve extract IconImage (need to call Extract first)
- void SetFile (const File &f)

  Set/Get File.

Protected Member Functions

- void ExtractIconImages ()
- void ExtractVeproIconImages ()

25.124.1 Detailed Description

IconImageFilter This filter will extract icons from a gdcm::File This filter will loop over all known sequence (public and private) that may contains an IconImage and retrieve them. The filter will fails with a value of false if no icon can be found Since it handle both public and private icon type, one should not assume the icon is in uncompress form, some private vendor store private icon in JPEG8/JPEG12.

Implementation details: This filter supports the following Icons:

- (0088,0200) Icon Image Sequence
- (0009,10,GEIIS) GE IIS Thumbnail Sequence
- (6003,10,GEMS_Ultrasound_ImageGroup_001) GEMS Image Thumbnail Sequence

- (0055,30,VEPRO VIF 3.0 DATA) Icon Data
- (0055,30,VEPRO VIM 5.0 DATA) ICONDATA2

Warning

the icon stored in those private attribute do not conform to definition of Icon Image Sequence (do not simply copy/-paste). For example some private icon can be expressed as 12bits pixel, while the DICOM standard only allow 8bits icons.

See also

ImageReader

Examples:

ExtractIconFromFile.cxx.

25.124.2 Constructor & Destructor Documentation

25.124.2.1 gdcm::IconImageFilter::IconImageFilter ( )

25.124.2.2 gdcm::IconImageFilter::~IconImageFilter ( )

25.124.3 Member Function Documentation

25.124.3.1 bool gdcm::IconImageFilter::Extract ( )

Extract all Icon found in File.

Examples:

ExtractIconFromFile.cxx.

25.124.3.2 void gdcm::IconImageFilter::ExtractIconImages ( ) [protected]

25.124.3.3 void gdcm::IconImageFilter::ExtractVeproIconImages ( ) [protected]

25.124.3.4 File& gdcm::IconImageFilter::GetFile ( ) [inline]

25.124.3.5 const File& gdcm::IconImageFilter::GetFile ( ) const [inline]

25.124.3.6 IconImage& gdcm::IconImageFilter::GetIconImage (unsigned int *i*) const

Examples:

ExtractIconFromFile.cxx.

25.124.3.7 unsigned int gdcm::IconImageFilter::GetNumberOfIconImages ( ) const

Retrieve extract IconImage (need to call Extract first)

Examples:

ExtractIconFromFile.cxx.

25.124.3.8 void gdcm::IconImageFilter::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

ExtractIconFromFile.cxx.

The documentation for this class was generated from the following file:

- gdcmIconImageFilter.h

25.125 gdcm::IconImageGenerator Class Reference

IconImageGenerator This filter will generate a valid Icon from the Pixel Data element (an instance of gdcm::Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

```
#include <gdcmIconImageGenerator.h>
```

Public Member Functions

- IconImageGenerator ()
- ~IconImageGenerator ()
- void AutoPixelMinMax (bool b)
- void ConvertRGBToPaletteColor (bool b)
- bool Generate ()

  Generate Icon.
- const IconImage & GetIconImage () const

  Retrieve generated Icon.
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const
- void SetOutputDimensions (const unsigned int dims[2])

  Set Target dimension of output Icon.
- void SetOutsideValuePixel (double v)
- void SetPixelMinMax (double min, double max)
- void SetPixmap (const Pixmap &p)

  Set/Get File.

25.125.1 Detailed Description

IconImageGenerator This filter will generate a valid Icon from the Pixel Data element (an instance of gdcm::Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

- MONOCHROME1
- MONOCHROME2
- PALETTE_COLOR

The Pixel Bits Allocated is restricted to 8bits, therefore 16 bits image needs to be rescaled. By default the filter will use the full scalar range of 16bits image to rescale to unsigned 8bits. This may not be ideal for some situation, in which case the API SetPixelMinMax can be used to overwrite the default min,max interval used.

See also

ImageReader

Examples:

ExtractIconFromFile.cxx.

25.125.2 Constructor & Destructor Documentation

25.125.2.1 gdcm::IconImageGenerator::IconImageGenerator ( )

25.125.2.2 gdcm::IconImageGenerator::~IconImageGenerator ( )

25.125.3 Member Function Documentation

25.125.3.1 void gdcm::IconImageGenerator::AutoPixelMinMax (bool *b*)

Instead of explicitely specifying the min/max value for the rescale operation, let the internal mechanism compute the min/max of icon and rescale to best appropriate.

Examples:

ExtractIconFromFile.cxx.

25.125.3.2 void gdcm::IconImageGenerator::ConvertRGBToPaletteColor (bool *b*)

Converting from RGB to PALETTE_COLOR can be a slow operation. However DICOM standard requires that color icon be described as palette. Set this boolean to false only if you understand the consequences. true, false generates invalid Icon Image Sequence

25.125.3.3 bool gdcm::IconImageGenerator::Generate ( )

Generate Icon.

Examples:

ExtractIconFromFile.cxx.

25.125.3.4 const IconImage& gdcm::IconImageGenerator::GetIconImage ( ) const [inline]

Retrieve generated Icon.

Examples:

ExtractIconFromFile.cxx.

25.125.3.5 Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) [inline]

25.125.3.6 const Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) const [inline]

25.125.3.7 void gdcm::IconImageGenerator::SetOutputDimensions (const unsigned int *dims[2]*)

Set Target dimension of output Icon.

Examples:

ExtractIconFromFile.cxx.

25.125.3.8 void gdcm::IconImageGenerator::SetOutsideValuePixel (double *v*)

Set a pixel value that should be discarded. This happen typically for CT image, where a pixel has been used to pad outside the image (see Pixel Padding Value). Requires AutoPixelMinMax(true)

25.125.3.9 void gdcm::IconImageGenerator::SetPixelMinMax (double *min,* double *max*)

Override default min/max to compute best rescale for 16bits -> 8bits downscale. Typically those value can be read from the SmallestImagePixelValue LargestImagePixelValue DICOM attribute.

25.125.3.10 void gdcm::IconImageGenerator::SetPixmap (const Pixmap & *p*) [inline]

Set/Get File.

Examples:

ExtractIconFromFile.cxx.

The documentation for this class was generated from the following file:

- gdcmIconImageGenerator.h

25.126 gdcm::ignore_char Struct Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- ignore_char (char c)

Public Attributes

- char m_char

25.126.1 Constructor & Destructor Documentation

25.126.1.1 gdcm::ignore_char::ignore_char (char *c*) [inline]

25.126.2 Member Data Documentation

25.126.2.1 char gdcm::ignore_char::m_char

Referenced by gdcm::operator>>().

The documentation for this struct was generated from the following file:

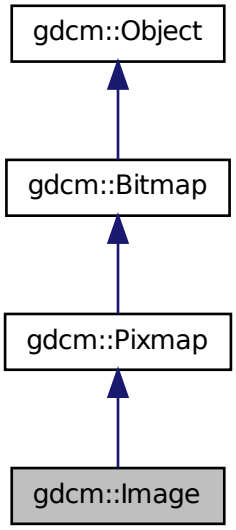
- gdcmElement.h

25.127 gdcm::Image Class Reference

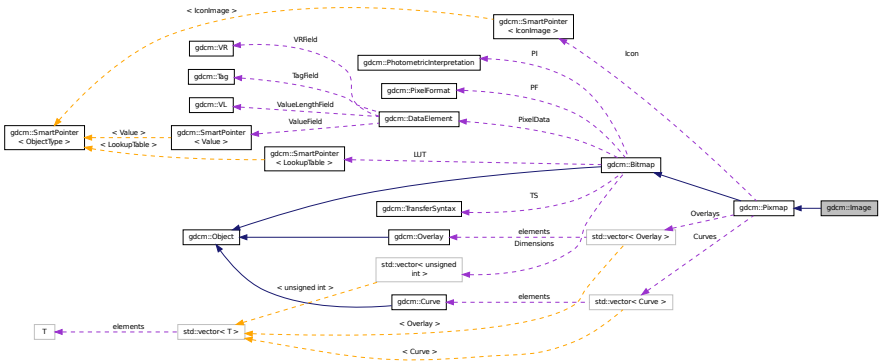
Image.

```
#include <gdcmImage.h>
```

Inheritance diagram for gdcm::Image:



Collaboration diagram for gdcm::Image:



Public Member Functions

- Image ()
- ∼Image ()
- const double ∗ GetDirectionCosines () const
- double GetDirectionCosines (unsigned int idx) const
- double GetIntercept () const
- const double ∗ GetOrigin () const
- double GetOrigin (unsigned int idx) const
- double GetSlope () const
- const double ∗ GetSpacing () const
- double GetSpacing (unsigned int idx) const
- void Print (std::ostream &os) const

  print
- void SetDirectionCosines (const float ∗dircos)
- void SetDirectionCosines (const double ∗dircos)
- void SetDirectionCosines (unsigned int idx, double dircos)
- void SetIntercept (double intercept)

  intercept
- void SetOrigin (const float ∗ori)
- void SetOrigin (const double ∗ori)
- void SetOrigin (unsigned int idx, double ori)
- void SetSlope (double slope)

  slope
- void SetSpacing (const double ∗spacing)
- void SetSpacing (unsigned int idx, double spacing)

Additional Inherited Members

25.127.1 Detailed Description

Image.

Note

This is the container for an Image in the general sense. From this container you should be able to request information like:

- Origin
- Dimension
- PixelFormat ... But also to retrieve the image as a raw buffer (char ∗) Since we have to deal with both RAW data and JPEG stream (which internally encode all the above information) this API might seems redundant. One way to solve that would be to subclass gdcm::Image with gdcm::JPEGImage which would from the stream extract the header info and fill it to please gdcm::Image...well except origin for instance

Basically you can see it as a storage for the Pixel Data element (7fe0,0010).

Warning

This class does some heuristics to guess the Spacing but is not compatible with DICOM CP-586. In case of doubt use PixmapReader instead

See also

ImageReader PixmapReader

Examples:

CompressImage.cxx, ConvertToQImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GetJPEGSamplePrecision.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

25.127.2 Constructor & Destructor Documentation

25.127.2.1 gdcm::Image::Image ( ) [inline]

25.127.2.2 gdcm::Image::∼Image ( ) [inline]

25.127.3 Member Function Documentation

25.127.3.1 const double∗ gdcm::Image::GetDirectionCosines ( ) const

Return a 6-tuples specifying the direction cosines A default value of (1,0,0,0,1,0) will be return when the direction cosines was not specified.

25.127.3.2 double gdcm::Image::GetDirectionCosines (unsigned int *idx*) const

25.127.3.3 double gdcm::Image::GetIntercept ( ) const [inline]

25.127.3.4 const double∗ gdcm::Image::GetOrigin ( ) const

Return a 3-tuples specifying the origin Will return (0,0,0) if the origin was not specified.

Examples:

HelloVizWorld.cxx.

25.127.3.5 double gdcm::Image::GetOrigin (unsigned int *idx*) const

25.127.3.6 double gdcm::Image::GetSlope ( ) const [inline]

25.127.3.7 const double* gdcm::Image::GetSpacing ( ) const

Return a 3-tuples specifying the spacing NOTE: 3rd value can be an aribtrary 1 value when the spacing was not specified (ex. 2D image). WARNING: when the spacing is not specifier, a default value of 1 will be returned

25.127.3.8 double gdcm::Image::GetSpacing (unsigned int *idx*) const

25.127.3.9 void gdcm::Image::Print (std::ostream & *os*) const [virtual]

print

Reimplemented from gdcm::Pixmap.

Examples:

CompressImage.cxx, and PatchFile.cxx.

25.127.3.10 void gdcm::Image::SetDirectionCosines (const float * *dircos*)

25.127.3.11 void gdcm::Image::SetDirectionCosines (const double * *dircos*)

25.127.3.12 void gdcm::Image::SetDirectionCosines (unsigned int *idx,* double *dircos*)

25.127.3.13 void gdcm::Image::SetIntercept (double *intercept*) [inline]

intercept

25.127.3.14 void gdcm::Image::SetOrigin (const float * *ori*)

25.127.3.15 void gdcm::Image::SetOrigin (const double * *ori*)

25.127.3.16 void gdcm::Image::SetOrigin (unsigned int *idx,* double *ori*)

25.127.3.17 void gdcm::Image::SetSlope (double *slope*) [inline]

slope

25.127.3.18 void gdcm::Image::SetSpacing (const double * *spacing*)

Examples:

csa2img.cxx, and iU22tomultisc.cxx.

25.127.3.19 void gdcm::Image::SetSpacing (unsigned int *idx*, double *spacing*)

The documentation for this class was generated from the following file:

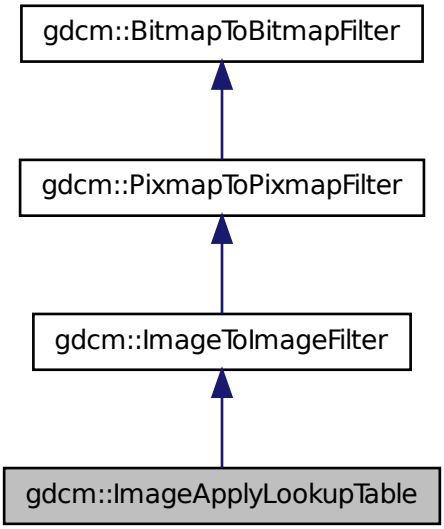
- gdcmImage.h

25.128 gdcm::ImageApplyLookupTable Class Reference

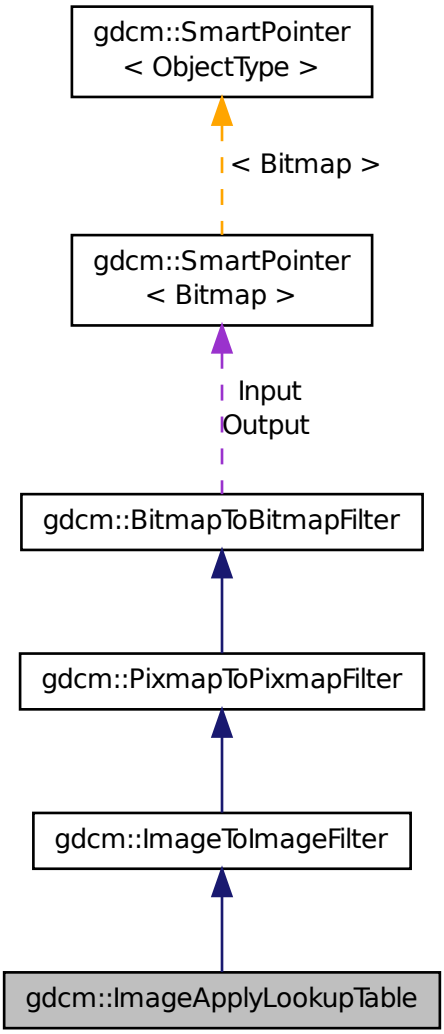
ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for gdcm::ImageApplyLookupTable:



Collaboration diagram for gdcm::ImageApplyLookupTable:



Public Member Functions

- ImageApplyLookupTable ()
- ~ImageApplyLookupTable ()
- bool Apply ()

  Apply.

25.128.1 Detailed Description

ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

25.128.2 Constructor & Destructor Documentation

25.128.2.1 gdcm::ImageApplyLookupTable::ImageApplyLookupTable ( ) [inline]

25.128.2.2 gdcm::ImageApplyLookupTable::~ImageApplyLookupTable ( ) [inline]

25.128.3 Member Function Documentation

25.128.3.1 bool gdcm::ImageApplyLookupTable::Apply ( )

Apply.

The documentation for this class was generated from the following file:

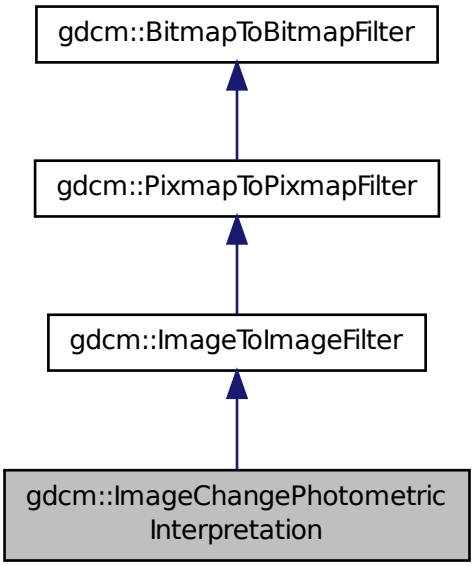
- gdcmImageApplyLookupTable.h

25.129 gdcm::ImageChangePhotometricInterpretation Class Reference

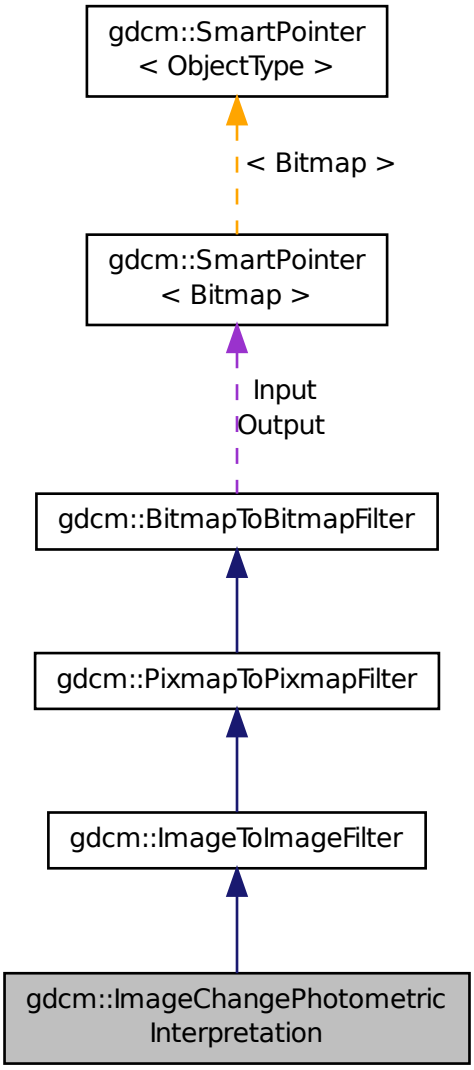
ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.

```
#include <gdcmImageChangePhotometricInterpretation.h>
```

Inheritance diagram for gdcm::ImageChangePhotometricInterpretation:



Collaboration diagram for gdcm::ImageChangePhotometricInterpretation:



Public Member Functions

- ImageChangePhotometricInterpretation ()
- ~ImageChangePhotometricInterpretation ()
- bool Change ()

  Change.
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)

  Set/Get requested PhotometricInterpretation.

Static Public Member Functions

- template<typename T >
  static void RGB2YBR (T ybr[3], const T rgb[3])

  colorspace converstion (based on CCIR Recommendation 601-2)
- template<typename T >
  static void YBR2RGB (T rgb[3], const T ybr[3])

Protected Member Functions

- bool ChangeMonochrome ()

25.129.1 Detailed Description

ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.

25.129.2 Constructor & Destructor Documentation

25.129.2.1 gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ( ) [inline]

25.129.2.2 gdcm::ImageChangePhotometricInterpretation::~ImageChangePhotometricInterpretation ( ) [inline]

25.129.3 Member Function Documentation

25.129.3.1 bool gdcm::ImageChangePhotometricInterpretation::Change ( )

Change.

25.129.3.2 bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ( ) [protected]

25.129.3.3 const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometricInterpretation ( ) const [inline]

25.129.3.4 template<typename T > void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (T *ybr[3]*, const T *rgb[3]*) [static]

colorspace converstion (based on CCIR Recommendation 601-2)

25.129.3.5 void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (PhotometricInterpretation const & *pi*) [inline]

Set/Get requested PhotometricInterpretation.

25.129.3.6 template<typename T > void gdcm::ImageChangePhotometricInterpretation::YBR2RGB (T *rgb[3]*, const T *ybr[3]*) [static]

The documentation for this class was generated from the following file:

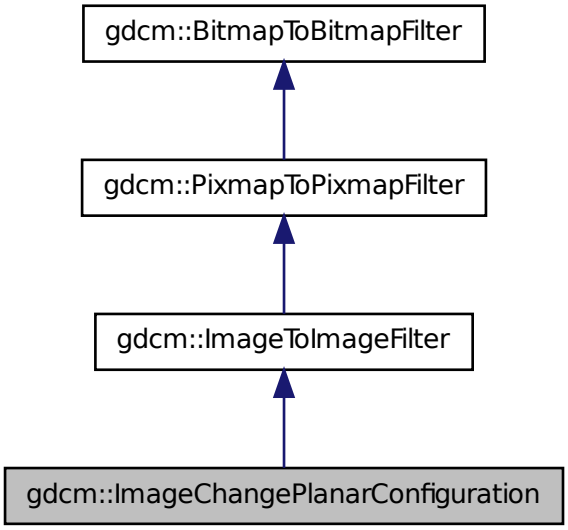
- gdcmImageChangePhotometricInterpretation.h

25.130 gdcm::ImageChangePlanarConfiguration Class Reference

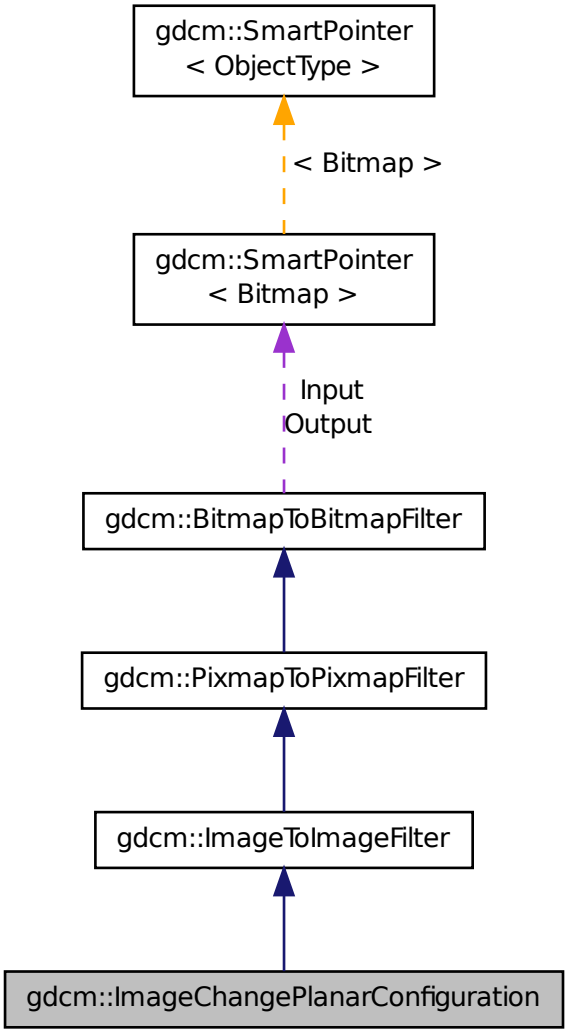
ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.

```
#include <gdcmImageChangePlanarConfiguration.h>
```

Inheritance diagram for gdcm::ImageChangePlanarConfiguration:



Collaboration diagram for gdcm::ImageChangePlanarConfiguration:



Public Member Functions

- ImageChangePlanarConfiguration ()
- ~ImageChangePlanarConfiguration ()
- bool Change ()

  Change.
- unsigned int GetPlanarConfiguration () const
- void SetPlanarConfiguration (unsigned int pc)

  Set/Get requested PlanarConfigation.

Static Public Member Functions

- template<typename T >
  static size_t RGBPixelsToRGBPlanes (T *r, T *g, T *b, const T *rgb, size_t s)
- template<typename T >
  static size_t RGBPlanesToRGBPixels (T *out, const T *r, const T *g, const T *b, size_t s)

25.130.1 Detailed Description

ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.

25.130.2 Constructor & Destructor Documentation

25.130.2.1 gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ( ) [inline]

25.130.2.2 gdcm::ImageChangePlanarConfiguration::~ImageChangePlanarConfiguration ( ) [inline]

25.130.3 Member Function Documentation

25.130.3.1 bool gdcm::ImageChangePlanarConfiguration::Change ( )

Change.

25.130.3.2 unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration ( ) const [inline]

25.130.3.3 template<typename T > size_t gdcm::ImageChangePlanarConfiguration::RGBPixelsToRGBPlanes (T * *r*, T * *g*, T * *b*, const T * *rgb*, size_t *s*) [static]

Convert a regular RGB pixel image (R,G,B,R,G,B...) into a planar R,G,B image (R,R..,G,G...B,B)

Warning

this works on a frame basis, you need to loop over all frames in multiple frames image to apply this function

25.130.3.4 template<typename T > size_t gdcm::ImageChangePlanarConfiguration::RGBPlanesToRGBPixels (T * *out*, const T * *r*, const T * *g*, const T * *b*, size_t *s*) [static]

s is the size of one plane (r,g or b). Thus the output buffer needs to be at least 3*s bytes long s can be seen as the number of RGB pixels in the output

25.130.3.5 void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (unsigned int *pc*) [inline]

Set/Get requested PlanarConfigation.

The documentation for this class was generated from the following file:

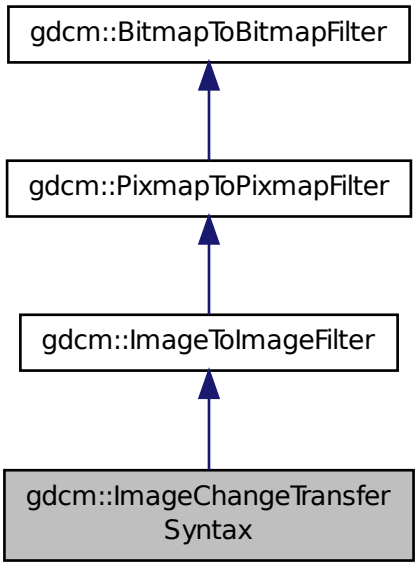
- gdcmImageChangePlanarConfiguration.h

25.131 gdcm::ImageChangeTransferSyntax Class Reference

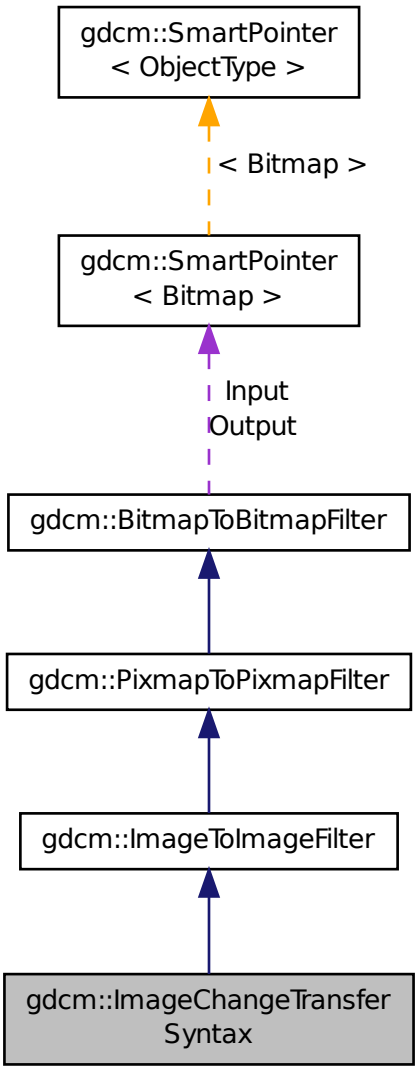
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.

```
#include <gdcmImageChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::ImageChangeTransferSyntax:



Collaboration diagram for gdcm::ImageChangeTransferSyntax:



Public Member Functions

- ImageChangeTransferSyntax ()
- ~ImageChangeTransferSyntax ()
- bool Change ()

  Change.
- const TransferSyntax & GetTransferSyntax () const

  Get Transfer Syntax.
- void SetCompressIconImage (bool b)

- void SetForce (bool f)
- void SetTransferSyntax (const TransferSyntax &ts)

  Set target Transfer Syntax.
- void SetUserCodec (ImageCodec *ic)

Protected Member Functions

- bool TryJPEG2000Codec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryJPEGCodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryJPEGLSCodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryRAWCodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryRLECodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)

25.131.1 Detailed Description

ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.

If only Force param is set but no input TransferSyntax is set, it is assumed that user only wants to inspect encapsulated stream (advanced dev. option).

When using UserCodec it is very important that the TransferSyntax (as set in SetTransferSyntax) is actually understood by UserCodec (ie. UserCodec->CanCode(TransferSyntax)). Otherwise the behavior is to use a default codec.

See also

JPEGCodec JPEGLSCodec JPEG2000Codec

Examples:

CompressImage.cxx.

25.131.2 Constructor & Destructor Documentation

25.131.2.1 gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax ( ) [inline]

25.131.2.2 gdcm::ImageChangeTransferSyntax::~ImageChangeTransferSyntax ( ) [inline]

25.131.3 Member Function Documentation

25.131.3.1 bool gdcm::ImageChangeTransferSyntax::Change ( )

Change.

Examples:

CompressImage.cxx.

25.131.3.2 const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax ( ) const [inline]

Get Transfer Syntax.

25.131.3.3 void gdcm::ImageChangeTransferSyntax::SetCompressIconImage (bool *b*) [inline]

Decide whether or not to also compress the Icon Image using the same Transfer Syntax Default is to simply decompress icon image

25.131.3.4 void gdcm::ImageChangeTransferSyntax::SetForce (bool *f*) [inline]

When target Transfer Syntax is identical to input target syntax, no operation is actually done This is an issue when someone wants to recompress using GDCM internal implementation a JPEG (for example) image

25.131.3.5 void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (const TransferSyntax & *ts*) [inline]

Set target Transfer Syntax.

Examples:

CompressImage.cxx.

25.131.3.6 void gdcm::ImageChangeTransferSyntax::SetUserCodec (ImageCodec * *ic*) [inline]

Allow user to specify exactly which codec to use. this is needed to specify special qualities or compression option.

Warning

is the codec 'ic' is not compatible with the TransferSyntax requested, it will not be used. It is the user responsability to check that UserCodec->CanCode(TransferSyntax)

25.131.3.7 bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

25.131.3.8 bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

25.131.3.9 bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

25.131.3.10 bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

25.131.3.11 bool gdcm::ImageChangeTransferSyntax::TryRLECodec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

The documentation for this class was generated from the following file:

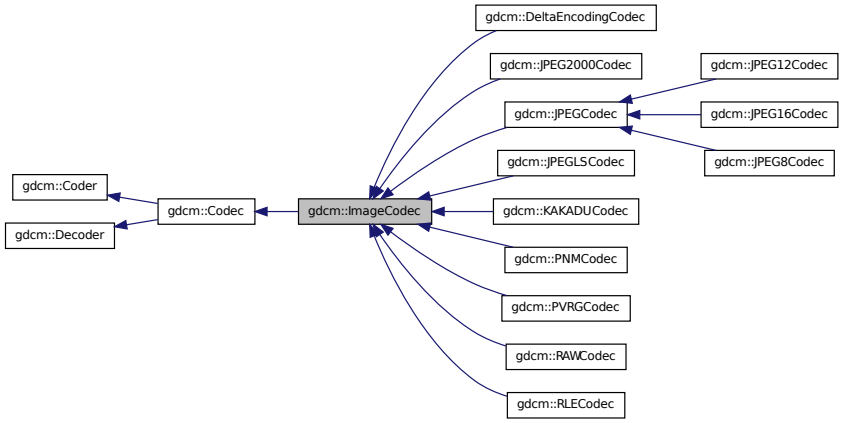
- gdcmImageChangeTransferSyntax.h

25.132 gdcm::ImageCodec Class Reference

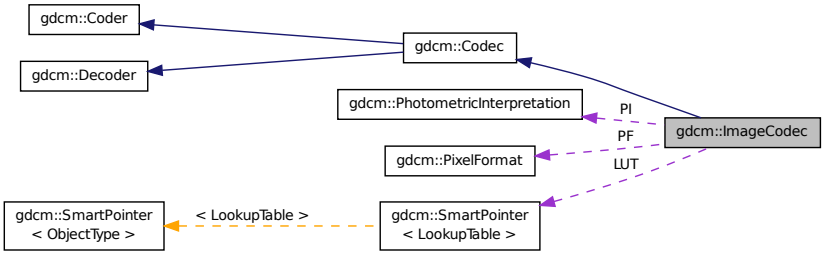
ImageCodec.

`#include <gdcmImageCodec.h>`

Inheritance diagram for gdcm::ImageCodec:



Collaboration diagram for gdcm::ImageCodec:



Public Member Functions

- ImageCodec ()
- ~ImageCodec ()
- bool CanCode (TransferSyntax const &) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Decode (DataElement const &is_, DataElement &os)

  Decode.
- const unsigned int * GetDimensions () const

- virtual bool GetHeaderInfo (std::istream &is_, TransferSyntax &ts)
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const unsigned int *d)
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Protected Types

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions

- bool Decode (std::istream &is_, std::ostream &os)
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- virtual bool IsValid (PhotometricInterpretation const &pi)

Protected Attributes

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

Friends

- class ImageChangePhotometricInterpretation

25.132.1 Detailed Description

ImageCodec.

Note

Main codec, this is a central place for all implementation

25.132.2 Member Typedef Documentation

25.132.2.1 typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr [protected]

25.132.3 Constructor & Destructor Documentation

25.132.3.1 gdcm::ImageCodec::ImageCodec ( )

25.132.3.2 gdcm::ImageCodec::~ImageCodec ( )

25.132.4 Member Function Documentation

25.132.4.1 bool gdcm::ImageCodec::CanCode (TransferSyntax const &) const [inline], [virtual]

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

Reimplemented in gdcm::JPEGCodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::PNMCodec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

25.132.4.2 bool gdcm::ImageCodec::CanDecode (TransferSyntax const &) const [inline], [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

Reimplemented in gdcm::JPEGCodec, gdcm::PVRGCodec, gdcm::RLECodec, gdcm::JPEG2000Codec, gdcm::JPEGLSCodec, gdcm::PNMCodec, gdcm::RAWCodec, and gdcm::KAKADUCodec.

25.132.4.3 bool gdcm::ImageCodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEGCodec, gdcm::PVRGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::DeltaEncodingCodec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

25.132.4.4 bool gdcm::ImageCodec::Decode (std::istream & *is_*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEGCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::RAWCodec, gdcm::DeltaEncodingCodec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

25.132.4.5 bool gdcm::ImageCodec::DoByteSwap (std::istream & *is_*, std::ostream & *os*) [protected]

25.132.4.6 bool gdcm::ImageCodec::DoInvertMonochrome (std::istream & *is_*, std::ostream & *os*) [protected]

25.132.4.7 bool gdcm::ImageCodec::DoOverlayCleanup (std::istream & *is_*, std::ostream & *os*) [protected]

25.132.4.8 bool gdcm::ImageCodec::DoPaddedCompositePixelCode (std::istream & *is_*, std::ostream & *os*) [protected]

25.132.4.9 bool gdcm::ImageCodec::DoPlanarConfiguration (std::istream & *is_*, std::ostream & *os*) [protected]

25.132.4.10 bool gdcm::ImageCodec::DoSimpleCopy (std::istream & *is_*, std::ostream & *os*) [protected]

25.132.4.11 bool gdcm::ImageCodec::DoYBR (std::istream & *is_*, std::ostream & *os*) [protected]

25.132.4.12 const unsigned int* gdcm::ImageCodec::GetDimensions ( ) const [inline]

25.132.4.13 virtual bool gdcm::ImageCodec::GetHeaderInfo (std::istream & *is_*, TransferSyntax & *ts*) [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::RLECodec, gdcm::JPEG2000Codec, gdcm::PNMCodec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, gdcm::JPEG8Codec, and gdcm::RAWCodec.

25.132.4.14 bool gdcm::ImageCodec::GetLossyFlag ( ) const

25.132.4.15 const LookupTable& gdcm::ImageCodec::GetLUT ( ) const [inline]

25.132.4.16 bool gdcm::ImageCodec::GetNeedByteSwap ( ) const [inline]

25.132.4.17 unsigned int gdcm::ImageCodec::GetNumberOfDimensions ( ) const

25.132.4.18 const PhotometricInterpretation& gdcm::ImageCodec::GetPhotometricInterpretation ( ) const

25.132.4.19 PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) [inline]

Examples:

GetJPEGSamplePrecision.cxx.

25.132.4.20 const PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) const [inline]

25.132.4.21 unsigned int gdcm::ImageCodec::GetPlanarConfiguration ( ) const [inline]

25.132.4.22 bool gdcm::ImageCodec::IsLossy ( ) const

25.132.4.23 virtual bool gdcm::ImageCodec::IsValid (PhotometricInterpretation const & *pi*) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec.

25.132.4.24 void gdcm::ImageCodec::SetDimensions (const unsigned int * *d*) [inline]

Examples:

ExtractIconFromFile.cxx.

25.132.4.25 void gdcm::ImageCodec::SetDimensions (const std::vector< unsigned int > & *d*) [inline]

25.132.4.26 void gdcm::ImageCodec::SetLossyFlag (bool *l*)

25.132.4.27 void gdcm::ImageCodec::SetLUT (LookupTable const & *lut*) [inline]

Examples:

ExtractIconFromFile.cxx.

25.132.4.28 void gdcm::ImageCodec::SetNeedByteSwap (bool *b*) [inline]

25.132.4.29 void gdcm::ImageCodec::SetNeedOverlayCleanup (bool *b*) [inline]

25.132.4.30 void gdcm::ImageCodec::SetNumberOfDimensions (unsigned int *dim*)

25.132.4.31 void gdcm::ImageCodec::SetPhotometricInterpretation (PhotometricInterpretation const & *pi*)

Examples:

ExtractIconFromFile.cxx.

25.132.4.32 virtual void gdcm::ImageCodec::SetPixelFormat (PixelFormat const & *pf*) [inline], [virtual]

Reimplemented in gdcm::JPEGCodec.

Examples:

ExtractIconFromFile.cxx.

25.132.4.33 void gdcm::ImageCodec::SetPlanarConfiguration (unsigned int *pc*) [inline]

25.132.5 Friends And Related Function Documentation

25.132.5.1 friend class ImageChangePhotometricInterpretation [friend]

25.132.6 Member Data Documentation

25.132.6.1 unsigned int gdcm::ImageCodec::Dimensions[3] [protected]

25.132.6.2 bool gdcm::ImageCodec::LossyFlag [protected]

25.132.6.3 LUTPtr gdcm::ImageCodec::LUT [protected]

25.132.6.4 bool gdcm::ImageCodec::NeedByteSwap [protected]

25.132.6.5 bool gdcm::ImageCodec::NeedOverlayCleanup [protected]

25.132.6.6 unsigned int gdcm::ImageCodec::NumberOfDimensions [protected]

25.132.6.7 PixelFormat gdcm::ImageCodec::PF [protected]

25.132.6.8 PhotometricInterpretation gdcm::ImageCodec::PI [protected]

25.132.6.9 unsigned int gdcm::ImageCodec::PlanarConfiguration [protected]

25.132.6.10 bool gdcm::ImageCodec::RequestPaddedCompositePixelCode [protected]

25.132.6.11 bool gdcm::ImageCodec::RequestPlanarConfiguration [protected]

The documentation for this class was generated from the following file:

- gdcmImageCodec.h

25.133 gdcm::ImageConverter Class Reference

Image Converter.

`#include <gdcmImageConverter.h>`

Public Member Functions

- ImageConverter ()
- ~ImageConverter ()
- void Convert ()
- const Image & GetOuput () const
- void SetInput (Image const &input)

25.133.1 Detailed Description

Image Converter.

Note

This is the class used to convert from on gdcm::Image to another This is typically used to convert let say YBR JPEG compressed gdcm::Image to a RAW RGB gdcm::Image. So that the buffer can be directly pass to third party application. This filter is application level and not integrated directly in GDCM

25.133.2 Constructor & Destructor Documentation

25.133.2.1 gdcm::ImageConverter::ImageConverter ( )

25.133.2.2 gdcm::ImageConverter::~ImageConverter ( )

25.133.3 Member Function Documentation

25.133.3.1 void gdcm::ImageConverter::Convert ( )

25.133.3.2 const Image& gdcm::ImageConverter::GetOuput ( ) const

25.133.3.3 void gdcm::ImageConverter::SetInput (Image const & *input*)

The documentation for this class was generated from the following file:

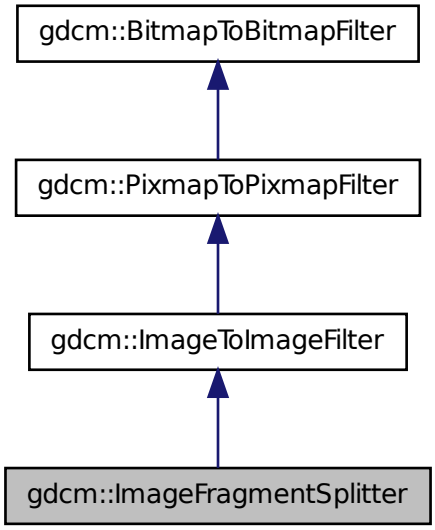
- gdcmImageConverter.h

25.134 gdcm::ImageFragmentSplitter Class Reference

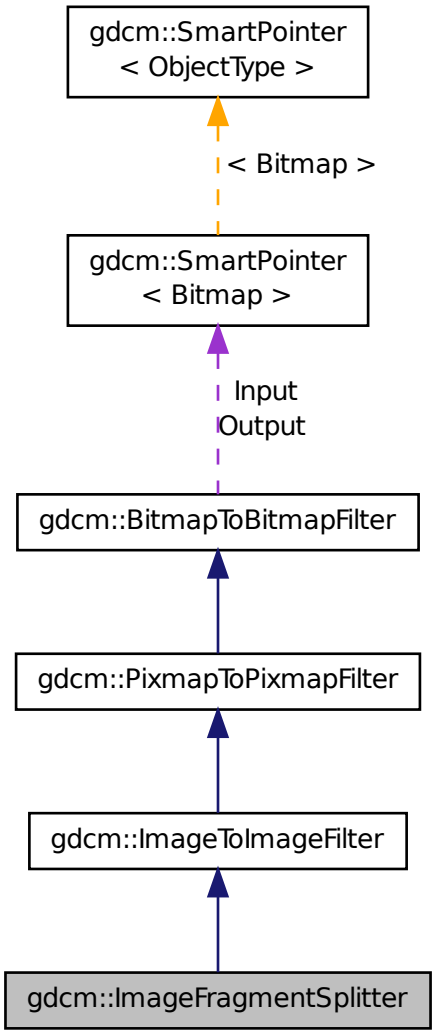
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

`#include <gdcmImageFragmentSplitter.h>`

Inheritance diagram for gdcm::ImageFragmentSplitter:



Collaboration diagram for gdcm::ImageFragmentSplitter:



Public Member Functions

- ImageFragmentSplitter ()
- ~ImageFragmentSplitter ()
- unsigned int GetFragmentSizeMax () const
- void SetForce (bool f)
- void SetFragmentSizeMax (unsigned int fragsize)

  FragmentSizeMax needs to be an even number.
- bool Split ()

  Split.

25.134.1 Detailed Description

ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

25.134.2 Constructor & Destructor Documentation

25.134.2.1 gdcm::ImageFragmentSplitter::ImageFragmentSplitter ( ) [inline]

25.134.2.2 gdcm::ImageFragmentSplitter::~ImageFragmentSplitter ( ) [inline]

25.134.3 Member Function Documentation

25.134.3.1 unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax ( ) const [inline]

25.134.3.2 void gdcm::ImageFragmentSplitter::SetForce (bool *f*) [inline]

When file already has all it's segment < FragmentSizeMax there is not need to run the filter. Unless the user explicitly say 'force' recomputation !

25.134.3.3 void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (unsigned int *fragsize*)

FragmentSizeMax needs to be an even number.

25.134.3.4 bool gdcm::ImageFragmentSplitter::Split ( )

Split.

The documentation for this class was generated from the following file:

- gdcmImageFragmentSplitter.h

25.135 gdcm::ImageHelper Class Reference

ImageHelper (internal class, not intended for user level)

```
#include <gdcmImageHelper.h>
```

Static Public Member Functions

- static bool ComputeSpacingFromImagePositionPatient (const std::vector< double > &imageposition, std::vector< double > &spacing)

  DO NOT USE.
- static std::vector< unsigned int > GetDimensionsValue (const File &f)
- static bool GetDirectionCosinesFromDataSet (DataSet const &ds, std::vector< double > &dircos)
- static std::vector< double > GetDirectionCosinesValue (File const &f)
- static bool GetForcePixelSpacing ()
- static bool GetForceRescaleInterceptSlope ()
- static SmartPointer< LookupTable > GetLUT (File const &f)
- static std::vector< double > GetOriginValue (File const &f)

Set/Get Origin (IPP) from/to a file.

- static PhotometricInterpretation GetPhotometricInterpretationValue (File const &f)
- static PixelFormat GetPixelFormatValue (const File &f)
- static unsigned int GetPlanarConfigurationValue (const File &f)
- static const ByteValue * GetPointerFromElement (Tag const &tag, File const &f)

Moved from PixampReader to here. Generally used for photometric interpretation.

- static std::vector< double > GetRescaleInterceptSlopeValue (File const &f)
- static std::vector< double > GetSpacingValue (File const &f)

Set/Get Spacing from/to a File.

- static void SetDimensionsValue (File &f, const Image &img)
- static void SetDirectionCosinesValue (DataSet &ds, const std::vector< double > &dircos)
- static void SetForcePixelSpacing (bool)
- static void SetForceRescaleInterceptSlope (bool)
- static void SetOriginValue (DataSet &ds, const Image &img)
- static void SetRescaleInterceptSlopeValue (File &f, const Image &img)
- static void SetSpacingValue (DataSet &ds, const std::vector< double > &spacing)

Static Protected Member Functions

- static Tag GetSpacingTagFromMediaStorage (MediaStorage const &ms)
- static Tag GetZSpacingTagFromMediaStorage (MediaStorage const &ms)

25.135.1 Detailed Description

ImageHelper (internal class, not intended for user level)

Helper for writing World images in DICOM. DICOM has a 'template' approach to image where MR Image Storage are distinct object from Enhanced MR Image Storage. For example the Pixel Spacing in one object is not at the same position (ie Tag) as in the other this class is the central (read: fragile) place where all the dispatching is done from a unified view of a world image (typically VTK or ITK point of view) down to the low level DICOM point of view.

Warning

: do not expect the API of this class to be maintained at any point, since as Modalities are added the API might have to be augmented or behavior changed to cope with new modalities.

25.135.2 Member Function Documentation

25.135.2.1 static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (const std::vector< double > & *imageposition,* std::vector< double > & *spacing*) [static]

DO NOT USE.

25.135.2.2 static std::vector<unsigned int> gdcm::ImageHelper::GetDimensionsValue (const File & *f*) [static]

This function checks tags (0x0028, 0x0010) and (0x0028, 0x0011) for the rows and columns of the image in pixels (as opposed to actual distances). The output is {col , row}

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.135.2.3 static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (DataSet const & *ds*, std::vector< double > & *dircos*) [static]

25.135.2.4 static std::vector<double> gdcm::ImageHelper::GetDirectionCosinesValue (File const & *f*) [static]

Get Direction Cosines (IOP) from/to a file Requires a file because mediastorage must be known

25.135.2.5 static bool gdcm::ImageHelper::GetForcePixelSpacing ( ) [static]

25.135.2.6 static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ( ) [static]

25.135.2.7 static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT (File const & *f*) [static]

25.135.2.8 static std::vector<double> gdcm::ImageHelper::GetOriginValue (File const & *f*) [static]

Set/Get Origin (IPP) from/to a file.

25.135.2.9 static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (File const & *f*) [static]

25.135.2.10 static PixelFormat gdcm::ImageHelper::GetPixelFormatValue (const File & *f*) [static]

This function returns pixel information about an image from its dataset That includes samples per pixel and bit depth (in that order)

25.135.2.11 static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (const File & *f*) [static]

25.135.2.12 static const ByteValue* gdcm::ImageHelper::GetPointerFromElement (Tag const & *tag*, File const & *f*) [static]

Moved from PixampReader to here. Generally used for photometric interpretation.

25.135.2.13 static std::vector<double> gdcm::ImageHelper::GetRescaleInterceptSlopeValue (File const & *f*) [static]

Set/Get shift/scale from/to a file

Warning

this function reads/sets the Slope/Intercept in appropriate class storage, but also Grid Scaling in RT Dose Storage Can't take a dataset because the mediastorage of the file must be known

25.135.2.14 static Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (MediaStorage const & *ms*) [static], [protected]

25.135.2.15 static std::vector<double> gdcm::ImageHelper::GetSpacingValue (File const & *f*) [static]

Set/Get Spacing from/to a File.

25.135.2.16 static Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (MediaStorage const & *ms*) [static], [protected]

25.135.2.17 static void gdcm::ImageHelper::SetDimensionsValue (File & *f*, const Image & *img*) [static]

25.135.2.18 static void gdcm::ImageHelper::SetDirectionCosinesValue (DataSet & *ds*, const std::vector< double > & *dircos*) [static]

Set Direction Cosines (IOP) from/to a file When IOD does not defines what is IOP (eg. typically Secondary Capture Image Storage) this call will simply remove the IOP attribute. Else in case of MR/CT image storage, this call will properly lookup the correct attribute to store the IOP.

25.135.2.19 static void gdcm::ImageHelper::SetForcePixelSpacing (bool) [static]

GDCM 1.x compatibility issue: When using ReWrite an MR Image Storage would be rewritten as Secondary Capture Object while still having a Pixel Spacing tag (0028,0030). If you have deal with those files, use this very special flag to handle them Unless explicitely set elsewhere by the standard, it will use value from 0028,0030 / 0018,0088 for the Pixel Spacing of the Image

25.135.2.20 static void gdcm::ImageHelper::SetForceRescaleInterceptSlope (bool) [static]

GDCM 1.x compatibility issue: when using ReWrite an MR Image Storage would be rewritten with a Rescale Slope/-Intercept while the standard would prohibit this (Philips Medical System is still doing that) Unless explicitely set elsewhere by the standard, it will use value from 0028,1052 / 0028,1053 for the Rescale Slope & Rescale Intercept values

25.135.2.21 static void gdcm::ImageHelper::SetOriginValue (DataSet & *ds*, const Image & *img*) [static]

25.135.2.22 static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (File & *f*, const Image & *img*) [static]

25.135.2.23 static void gdcm::ImageHelper::SetSpacingValue (DataSet & *ds*, const std::vector< double > & *spacing*) [static]

The documentation for this class was generated from the following file:

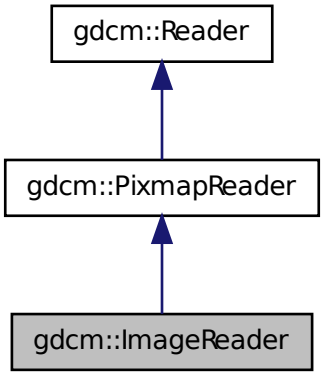
- gdcmImageHelper.h

25.136 gdcm::ImageReader Class Reference

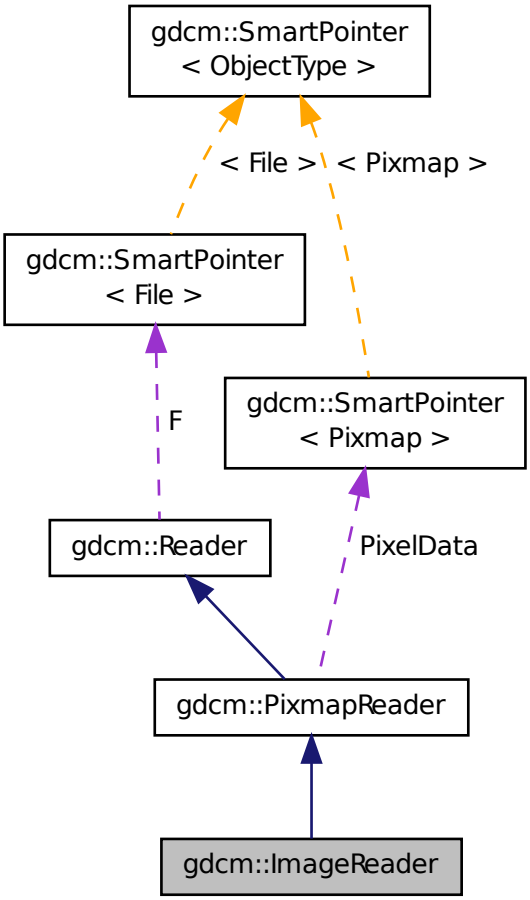
ImageReader.

```
#include <gdcmImageReader.h>
```

Inheritance diagram for gdcm::ImageReader:



Collaboration diagram for gdcm::ImageReader:



Public Member Functions

- ImageReader ()
- virtual ~ImageReader ()
- const Image & GetImage () const

  Return the read image.
- Image & GetImage ()
- virtual bool Read ()

Protected Member Functions

- bool ReadACRNEMAImage ()
- bool ReadImage (MediaStorage const &ms)

Additional Inherited Members

25.136.1 Detailed Description

ImageReader.

Note

its role is to convert the DICOM DataSet into a gdcm::Image representation Image is different from Pixmap has it has a position and a direction in Space.

See also

Image

Examples:

CheckBigEndianBug.cxx, CompressImage.cxx, ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.-cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MrProtocol.-cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

25.136.2 Constructor & Destructor Documentation

25.136.2.1 gdcm::ImageReader::ImageReader ( )

25.136.2.2 virtual gdcm::ImageReader::~ImageReader ( ) [virtual]

25.136.3 Member Function Documentation

25.136.3.1 const Image& gdcm::ImageReader::GetImage ( ) const

Return the read image.

Examples:

CompressImage.cxx, ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GetJPEG-SamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

25.136.3.2 Image& gdcm::ImageReader::GetImage ( )

25.136.3.3 virtual bool gdcm::ImageReader::Read ( ) [virtual]

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an Image.

Reimplemented from gdcm::PixmapReader.

Examples:

CheckBigEndianBug.cxx, CompressImage.cxx, ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

25.136.3.4 bool gdcm::ImageReader::ReadACRNEMAImage ( ) [protected], [virtual]

Reimplemented from gdcm::PixmapReader.

25.136.3.5 bool gdcm::ImageReader::ReadImage (MediaStorage const & *ms*) [protected], [virtual]

Reimplemented from gdcm::PixmapReader.

The documentation for this class was generated from the following file:

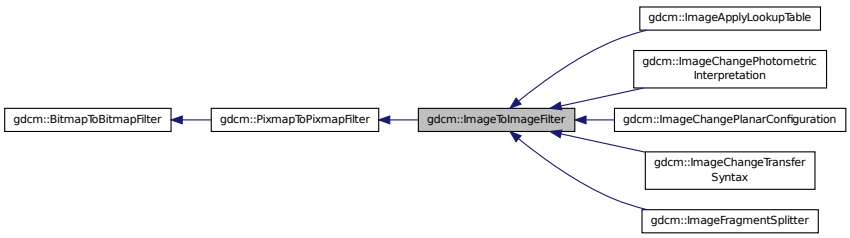
- gdcmImageReader.h

25.137 gdcm::ImageToImageFilter Class Reference

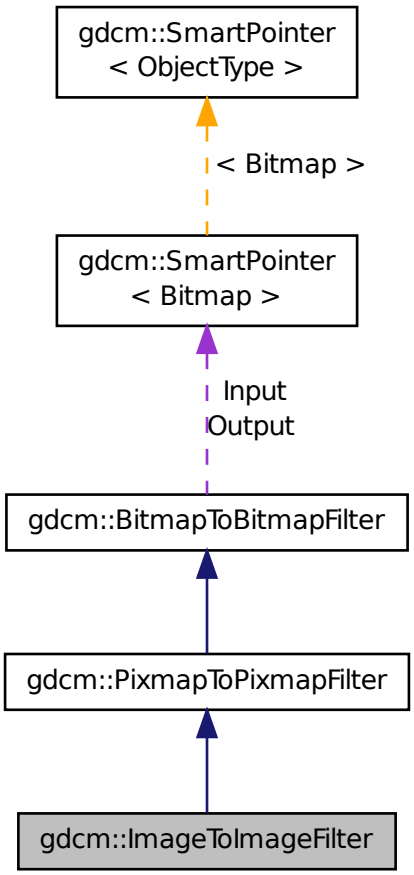
ImageToImageFilter class Super class for all filter taking an image and producing an output image.

`#include <gdcmImageToImageFilter.h>`

Inheritance diagram for gdcm::ImageToImageFilter:



Collaboration diagram for gdcm::ImageToImageFilter:



Public Member Functions

- ImageToImageFilter ()
- ~ImageToImageFilter ()
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

25.137.1 Detailed Description

ImageToImageFilter class Super class for all filter taking an image and producing an output image.

25.137.2 Constructor & Destructor Documentation

25.137.2.1 gdcm::ImageToImageFilter::ImageToImageFilter ( )

25.137.2.2 gdcm::ImageToImageFilter::~ImageToImageFilter ( ) [inline]

25.137.3 Member Function Documentation

25.137.3.1 Image& gdcm::ImageToImageFilter::GetInput ( )

Reimplemented from gdcm::PixmapToPixmapFilter.

25.137.3.2 const Image& gdcm::ImageToImageFilter::GetOutput ( ) const

Get Output image.

Reimplemented from gdcm::PixmapToPixmapFilter.

Examples:

CompressImage.cxx.

The documentation for this class was generated from the following file:

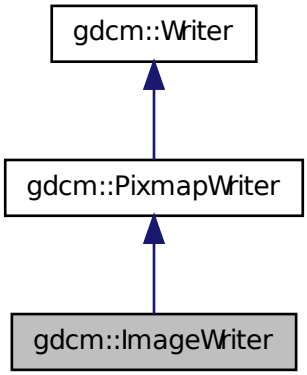
- gdcmImageToImageFilter.h

25.138 gdcm::ImageWriter Class Reference

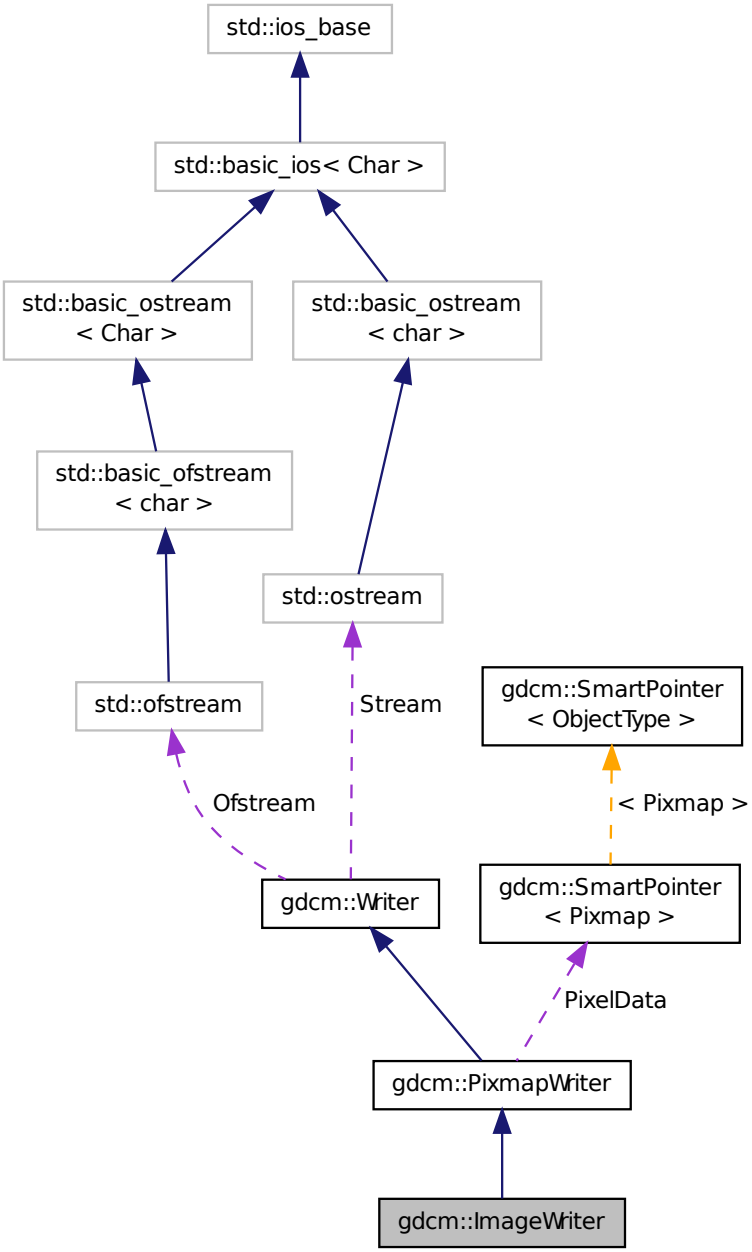
ImageWriter.

```
#include <gdcmImageWriter.h>
```

Inheritance diagram for gdcm::ImageWriter:



Collaboration diagram for gdcm::ImageWriter:



Public Member Functions

- ImageWriter ()
- ~ImageWriter ()
- const Image & GetImage () const

- Image & GetImage ()
- bool Write ()

  Write.

Additional Inherited Members

25.138.1 Detailed Description

ImageWriter.

Examples:

CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, and MergeTwoFiles.cxx.

25.138.2 Constructor & Destructor Documentation

25.138.2.1 gdcm::ImageWriter::ImageWriter ( )

25.138.2.2 gdcm::ImageWriter::∼ImageWriter ( )

25.138.3 Member Function Documentation

25.138.3.1 const Image& gdcm::ImageWriter::GetImage ( ) const [inline],[virtual]

Set/Get Image to be written It will overwrite anything Image infos found in DataSet (see parent class to see how to pass dataset)

Reimplemented from gdcm::PixmapWriter.

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

25.138.3.2 Image& gdcm::ImageWriter::GetImage ( ) [inline],[virtual]

Reimplemented from gdcm::PixmapWriter.

25.138.3.3 bool gdcm::ImageWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::PixmapWriter.

Examples:

CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, and MergeTwoFiles.cxx.

The documentation for this class was generated from the following file:

- gdcmImageWriter.h

25.139 gdcm::network::ImplementationClassUIDSub Class Reference

ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#include <gdcmImplementationClassUIDSub.h>
```

Public Member Functions

- ImplementationClassUIDSub ()
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.139.1 Detailed Description

ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

25.139.2 Constructor & Destructor Documentation

25.139.2.1 gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ( )

25.139.3 Member Function Documentation

25.139.3.1 std::istream& gdcm::network::ImplementationClassUIDSub::Read (std::istream & *is*)

25.139.3.2 size_t gdcm::network::ImplementationClassUIDSub::Size ( ) const

25.139.3.3 const std::ostream& gdcm::network::ImplementationClassUIDSub::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

- gdcmImplementationClassUIDSub.h

25.140 gdcm::network::ImplementationUIDSub Class Reference

ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

```
#include <gdcmImplementationUIDSub.h>
```

Public Member Functions

- ImplementationUIDSub ()
- const std::ostream & Write (std::ostream &os) const

25.140.1 Detailed Description

ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

25.140.2 Constructor & Destructor Documentation

25.140.2.1 gdcm::network::ImplementationUIDSub::ImplementationUIDSub ( )

25.140.3 Member Function Documentation

25.140.3.1 const std::ostream& gdcm::network::ImplementationUIDSub::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

- gdcmImplementationUIDSub.h

25.141 gdcm::network::ImplementationVersionNameSub Class Reference

ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#include <gdcmImplementationVersionNameSub.h>
```

Public Member Functions

- ImplementationVersionNameSub ()
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.141.1 Detailed Description

ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

25.141.2 Constructor & Destructor Documentation

25.141.2.1 gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ( )

25.141.3 Member Function Documentation

25.141.3.1 std::istream& gdcm::network::ImplementationVersionNameSub::Read (std::istream & *is*)

25.141.3.2 size_t gdcm::network::ImplementationVersionNameSub::Size ( ) const

25.141.3.3 const std::ostream& gdcm::network::ImplementationVersionNameSub::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

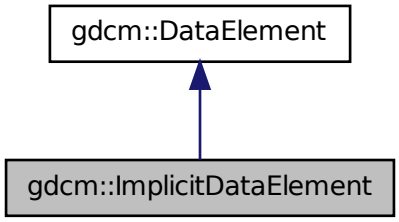
- gdcmImplementationVersionNameSub.h

25.142 gdcm::ImplicitDataElement Class Reference

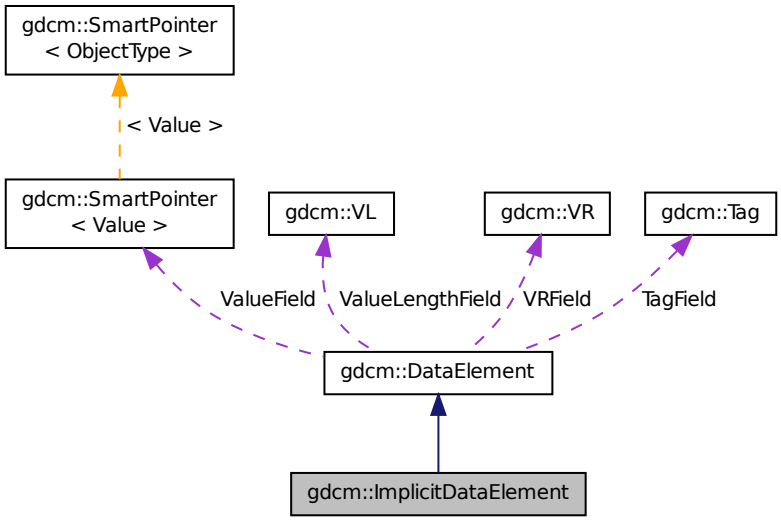
Class to represent an *Implicit VR* Data Element.

`#include <gdcmImplicitDataElement.h>`

Inheritance diagram for gdcm::ImplicitDataElement:



Collaboration diagram for gdcm::ImplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)

- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

25.142.1 Detailed Description

Class to represent an *Implicit VR* Data Element.

Note

bla

Examples:

ReadExplicitLengthSQIVR.cxx.

25.142.2 Member Function Documentation

25.142.2.1 VL gdcm::ImplicitDataElement::GetLength ( ) const

Reimplemented from gdcm::DataElement.

25.142.2.2 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::Read (std::istream & *is*)

Reimplemented from gdcm::DataElement.

25.142.2.3 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadPreValue (std::istream & *is*)

25.142.2.4 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValue (std::istream & *is*)

25.142.2.5 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadWithLength (std::istream & *is,* VL & *length*)

Reimplemented from gdcm::DataElement.

25.142.2.6 template<typename TSwap > const std::ostream& gdcm::ImplicitDataElement::Write (std::ostream & *os*) const

Reimplemented from gdcm::DataElement.

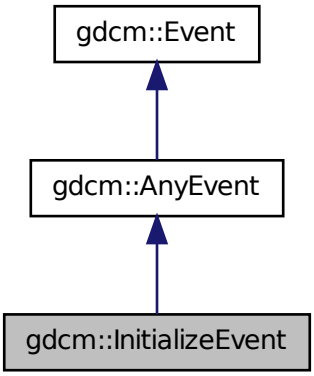
The documentation for this class was generated from the following file:

- gdcmImplicitDataElement.h

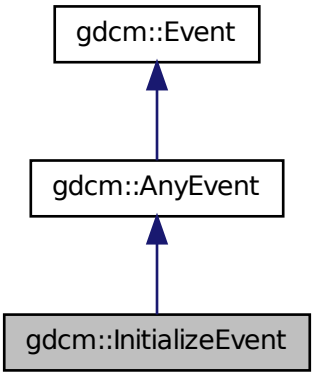
25.143 gdcm::InitializeEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::InitializeEvent:



Collaboration diagram for gdcm::InitializeEvent:



The documentation for this class was generated from the following file:

- gdcmEvent.h

25.144 gdcm::IOD Class Reference

Class for representing a IOD.

`#include <gdcmIOD.h>`

Public Types

- typedef std::vector< IODEntry > MapIODEntry
- typedef MapIODEntry::size_type SizeType

Public Member Functions

- IOD ()
- void AddIODEntry (const IODEntry &iode)
- void Clear ()
- const IODEntry & GetIODEntry (SizeType idx) const
- SizeType GetNumberOfIODs () const
- Type GetTypeFromTag (const Defs &defs, const Tag &tag) const

Friends

- std::ostream & operator<< (std::ostream &_os, const IOD &_val)

25.144.1 Detailed Description

Class for representing a IOD.

Note

bla

See also

Dict

Examples:

TraverseModules.cxx.

25.144.2 Member Typedef Documentation

25.144.2.1 typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry

25.144.2.2 typedef MapIODEntry::size_type gdcm::IOD::SizeType

25.144.3 Constructor & Destructor Documentation

25.144.3.1 gdcm::IOD::IOD ( ) [inline]

25.144.4 Member Function Documentation

25.144.4.1 void gdcm::IOD::AddIODEntry (const IODEntry & *iode*) [inline]

25.144.4.2 void gdcm::IOD::Clear ( ) [inline]

25.144.4.3 const IODEntry& gdcm::IOD::GetIODEntry (SizeType *idx*) const [inline]

Examples:

TraverseModules.cxx.

25.144.4.4 SizeType gdcm::IOD::GetNumberOfIODs ( ) const [inline]

Examples:

TraverseModules.cxx.

25.144.4.5 Type gdcm::IOD::GetTypeFromTag (const Defs & *defs,* const Tag & *tag*) const

25.144.5 Friends And Related Function Documentation

25.144.5.1 std::ostream& operator<< (std::ostream & *_os,* const IOD & *_val*) [friend]

The documentation for this class was generated from the following file:

- gdcmIOD.h

25.145 gdcm::IODEntry Class Reference

Class for representing a IODEntry.

```
#include <gdcmIODEntry.h>
```

Public Member Functions

- IODEntry (const char ∗name="", const char ∗ref="", const char ∗usag="")
- const char ∗ GetIE () const
- const char ∗ GetName () const
- const char ∗ GetRef () const
- const char ∗ GetUsage () const
- Usage::UsageType GetUsageType () const
- void SetIE (const char ∗ie)
- void SetName (const char ∗name)
- void SetRef (const char ∗ref)
- void SetUsage (const char ∗usag)

Friends

- std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)

25.145.1 Detailed Description

Class for representing a IODEntry.

Note

A.1.3 IOD Module Table and Functional Group Macro Table This Section of each IOD defines in a tabular form the Modules comprising the IOD. The following information must be specified for each Module in the table:

- The name of the Module or Functional Group
- A reference to the Section in Annex C which defines the Module or Functional Group
- The usage of the Module or Functional Group; whether it is:
- Mandatory (see A.1.3.1) , abbreviated M
- Conditional (see A.1.3.2) , abbreviated C
- User Option (see A.1.3.3) , abbreviated U The Modules referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each IOD, Mandatory Modules shall be supported per the definitions, semantics and requirements defined in Annex C. PS 3.3 - 2008 Page 96
- Standard - A.1.3.2 CONDITIONAL MODULES Conditional Modules are Mandatory Modules if specific conditions are met. If the specified conditions are not met, this Module shall not be supported; that is, no information defined in that Module shall be sent. A.1.3.3 USER OPTION MODULES User Option Modules may or may not be supported. If an optional Module is supported, the Attribute Types specified in the Modules in Annex C shall be supported.

See also

DictEntry

Examples:

TraverseModules.cxx.

25.145.2 Constructor & Destructor Documentation

25.145.2.1 gdcm::IODEntry::IODEntry (const char * *name* = " ", const char * *ref* = " ", const char * *usag* = " ") [inline]

25.145.3 Member Function Documentation

25.145.3.1 const char* gdcm::IODEntry::GetIE ( ) const [inline]

25.145.3.2 const char* gdcm::IODEntry::GetName ( ) const [inline]

25.145.3.3 const char* gdcm::IODEntry::GetRef ( ) const [inline]

Examples:

TraverseModules.cxx.

25.145.3.4 const char* gdcm::IODEntry::GetUsage ( ) const [inline]

25.145.3.5 Usage::UsageType gdcm::IODEntry::GetUsageType ( ) const

25.145.3.6 void gdcm::IODEntry::SetIE (const char * *ie*) [inline]

25.145.3.7 void gdcm::IODEntry::SetName (const char * *name*) [inline]

25.145.3.8 void gdcm::IODEntry::SetRef (const char * *ref*) [inline]

25.145.3.9 void gdcm::IODEntry::SetUsage (const char * *usag*) [inline]

25.145.4 Friends And Related Function Documentation

25.145.4.1 std::ostream& operator<< (std::ostream & *_os,* const IODEntry & *_val*) [friend]

The documentation for this class was generated from the following file:

- gdcmIODEntry.h

25.146 gdcm::IODs Class Reference

Class for representing a IODs.

```
#include <gdcmIODs.h>
```

Public Types

- typedef std::map< IODName, IOD > IODMapType
- typedef IODMapType::const_iterator IODMapTypeConstIterator
- typedef std::string IODName

Public Member Functions

- IODs ()
- void AddIOD (const char *name, const IOD &module)
- IODMapTypeConstIterator Begin () const
- void Clear ()
- IODMapTypeConstIterator End () const
- const IOD & GetIOD (const char *name) const

Friends

- std::ostream & operator<< (std::ostream &_os, const IODs &_val)

25.146.1 Detailed Description

Class for representing a IODs.

Note

bla

See also

IOD

Examples:

TraverseModules.cxx.

25.146.2 Member Typedef Documentation

25.146.2.1 typedef std::map<IODName, IOD> gdcm::IODs::IODMapType

25.146.2.2 typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator

25.146.2.3 typedef std::string gdcm::IODs::IODName

25.146.3 Constructor & Destructor Documentation

25.146.3.1 gdcm::IODs::IODs ( ) [inline]

25.146.4 Member Function Documentation

25.146.4.1 void gdcm::IODs::AddIOD (const char * *name*, const IOD & *module*) [inline]

25.146.4.2 IODMapTypeConstIterator gdcm::IODs::Begin ( ) const [inline]

25.146.4.3 void gdcm::IODs::Clear ( ) [inline]

25.146.4.4 IODMapTypeConstIterator gdcm::IODs::End ( ) const [inline]

25.146.4.5 const IOD& gdcm::IODs::GetIOD (const char * *name*) const [inline]

25.146.5 Friends And Related Function Documentation

25.146.5.1 std::ostream& operator<< (std::ostream & *_os*, const IODs & *_val*) [friend]

The documentation for this class was generated from the following file:

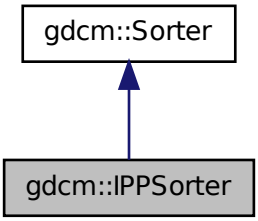
- gdcmIODs.h

25.147 gdcm::IPPSorter Class Reference

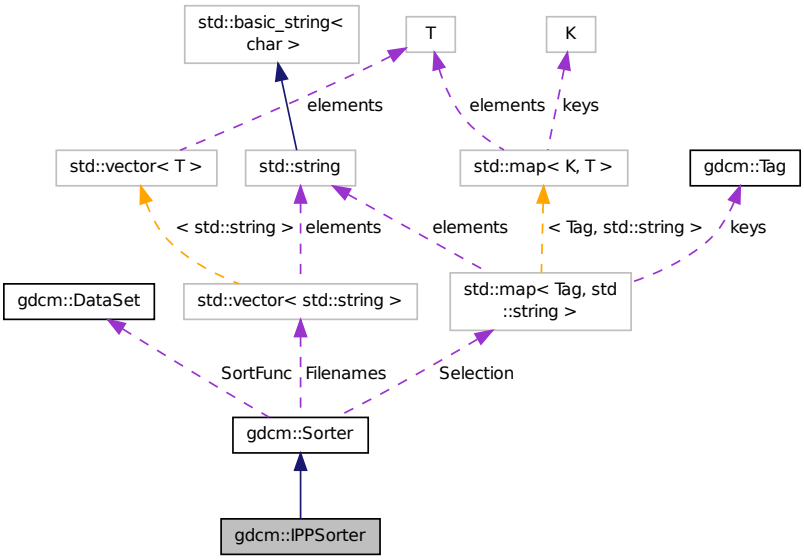
IPPSorter Implement a simple Image Position (Patient) sorter, along the Image Orientation (Patient) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

```
#include <gdcmIPPSorter.h>
```

Inheritance diagram for gdcm::IPPSorter:



Collaboration diagram for gdcm::IPPSorter:



Public Member Functions

- IPPSorter ()
- ~IPPSorter ()
- double GetDirectionCosinesTolerance () const
- double GetZSpacing () const
- double GetZSpacingTolerance () const
- void SetComputeZSpacing (bool b)

- void SetDirectionCosinesTolerance (double tol)
- void SetZSpacingTolerance (double tol)
- virtual bool Sort (std::vector< std::string > const &filenames)

Protected Attributes

- bool ComputeZSpacing
- double DirCosTolerance
- double ZSpacing
- double ZTolerance

Additional Inherited Members

25.147.1 Detailed Description

IPPSorter Implement a simple Image Position (Patient) sorter, along the Image Orientation (Patient) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

Warning

See special note for SetZSpacingTolerance when computing the ZSpacing from the IPP of each DICOM files (default tolerance for consistant spacing is: 1e-6mm)

For more information on Spacing, and how it is defined in DICOM, advanced users may refers to:

`http://sourceforge.net/apps/mediawiki/gdcm/index.php?title=Imager_Pixel_Spacing`

Bug There currently a couple of bug in this implementation:

- Frame Of Reference UID is not taken into account
- Gantry Tilt is not considered

Examples:

gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

25.147.2 Constructor & Destructor Documentation

25.147.2.1 gdcm::IPPSorter::IPPSorter ( )

25.147.2.2 gdcm::IPPSorter::~IPPSorter ( )

25.147.3 Member Function Documentation

25.147.3.1 double gdcm::IPPSorter::GetDirectionCosinesTolerance ( ) const [inline]

25.147.3.2 double gdcm::IPPSorter::GetZSpacing ( ) const [inline]

Read-only function to provide access to the computed value for the Z-Spacing The ComputeZSpacing must have been set to true before execution of sort algorithm. Call this function *after* calling Sort(); Z-Spacing will be 0 on 2 occasions:

- Sorting simply failed, potentially duplicate IPP => ZSpacing = 0
- ZSpacing could not be computed (Z-Spacing is not constant, or ZTolerance is too low)

Examples:

gdcmorthoplanes.cxx, and reslicesphere.cxx.

25.147.3.3 double gdcm::IPPSorter::GetZSpacingTolerance ( ) const [inline]

25.147.3.4 void gdcm::IPPSorter::SetComputeZSpacing (bool *b*) [inline]

Functions related to Z-Spacing computation Set to true when sort algorithm should also perform a regular Z-Spacing computation using the Image Position (Patient) Potential reason for failure:

1. ALL slices are taken into account, if one slice if missing then ZSpacing will be set to 0 since the spacing will not be found to be regular along the Series

Examples:

gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

25.147.3.5 void gdcm::IPPSorter::SetDirectionCosinesTolerance (double *tol*) [inline]

Sometimes IOP along a series is slightly changing for example: "0.999081\\0.0426953\\0.00369272\-0.0419025\\0.-955059\\0.293439", "0.999081\\0.0426953\\0.00369275\-0.0419025\\0.955059\\0.293439", "0.999081\\0.0426952\\0.-00369272\-0.0419025\\0.955059\\0.293439", We need an API to define the tolerance which is allowed. Internally the cross vector of each direction cosines is computed. The tolerance then define the the distance in between 1. to the dot product of those cross vectors. In a perfect world this dot product is of course 1.0 which imply a DirectionCosines tolerance of exactly 0.0 (default).

25.147.3.6 void gdcm::IPPSorter::SetZSpacingTolerance (double *tol*) [inline]

1. Another reason for failure is that that Z-Spacing is only slightly changing (eg 1e-3) along the serie, a human can determine that this is ok and change the tolerance from its default value: 1e-6

Examples:

gdcmorthoplanes.cxx, and reslicesphere.cxx.

25.147.3.7 virtual bool gdcm::IPPSorter::Sort (std::vector< std::string > const & *filenames*) [virtual]

Main entry point to the sorter. It will execute the filter, option should be set before running this function (SetZSpacing-Tolerance, ...) Return value indicate if sorting could be achived. Warning this does *NOT* imply that spacing is consistant, it only means the file are sorted according to IPP You should check if ZSpacing is 0 or not to deduce if file are actually a 3D volume

Reimplemented from gdcm::Sorter.

Examples:

gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

25.147.4 Member Data Documentation

25.147.4.1 bool gdcm::IPPSorter::ComputeZSpacing [protected]

25.147.4.2 double gdcm::IPPSorter::DirCosTolerance [protected]

25.147.4.3 double gdcm::IPPSorter::ZSpacing [protected]

25.147.4.4 double gdcm::IPPSorter::ZTolerance [protected]

The documentation for this class was generated from the following file:

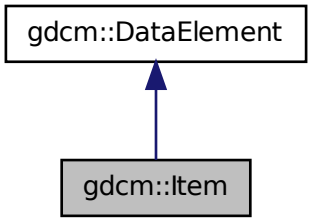
- gdcmIPPSorter.h

25.148 gdcm::Item Class Reference

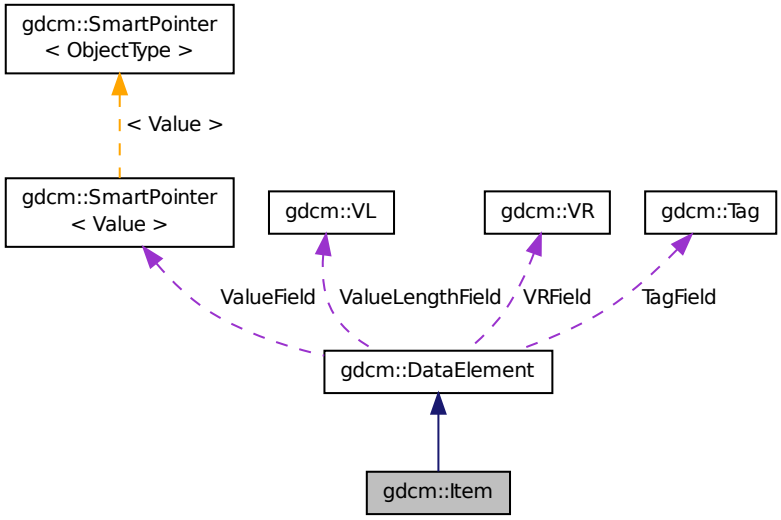
Class to represent an Item A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standart Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways Explicit/ Implicit.

```
#include <gdcmItem.h>
```

Inheritance diagram for gdcm::Item:



Collaboration diagram for gdcm::Item:



Public Member Functions

- Item ()
- Item (Item const &val)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR)
- bool FindDataElement (const Tag &t) const
- const DataElement & GetDataElement (const Tag &t) const
- template<typename TDE >
  VL GetLength () const
- const DataSet & GetNestedDataSet () const
- DataSet & GetNestedDataSet ()
- void InsertDataElement (const DataElement &de)
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is)
- void SetNestedDataSet (const DataSet &nested)
- template<typename TDE , typename TSwap >
  const std::ostream & Write (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &os, const Item &val)

Additional Inherited Members

25.148.1 Detailed Description

Class to represent an Item A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standart Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways Explicit/ Implicit.

Note

ITEM: A component of the Value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set.

Examples:

ChangeSequenceUltrasound.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, Extracting_All_-Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.-cxx, and LargeVRDSExplicit.cxx.

25.148.2 Constructor & Destructor Documentation

25.148.2.1 gdcm::Item::Item ( ) [inline]

25.148.2.2 gdcm::Item::Item (Item const & *val*) [inline]

25.148.3 Member Function Documentation

25.148.3.1 void gdcm::Item::Clear ( ) [inline]

Clear Data Element (make Value empty and invalidate Tag & VR)

Reimplemented from gdcm::DataElement.

References gdcm::DataElement::Clear().

Referenced by gdcm::SequenceOfItems::Read().

25.148.3.2 bool gdcm::Item::FindDataElement (const Tag & *t*) const [inline]

Examples:

ReadAndDumpDICOMDIR.cxx.

25.148.3.3 const DataElement& gdcm::Item::GetDataElement (const Tag & *t*) const [inline]

Examples:

ReadAndDumpDICOMDIR.cxx.

25.148.3.4 template<typename TDE > VL gdcm::Item::GetLength ( ) const

Reimplemented from gdcm::DataElement.

25.148.3.5 const DataSet& gdcm::Item::GetNestedDataSet ( ) const [inline]

Examples:

ChangeSequenceUltrasound.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, Extracting_All_-Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllV-R.cxx, GenFakeIdentifyFile.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, and LargeVRDSExplicit.cxx.

Referenced by gdcm::SequenceOfItems::Read().

25.148.3.6 DataSet& gdcm::Item::GetNestedDataSet ( ) [inline]

25.148.3.7 void gdcm::Item::InsertDataElement (const DataElement & *de*) [inline]

25.148.3.8 template<typename TDE , typename TSwap > std::istream& gdcm::Item::Read (std::istream & *is*) [inline]

Reimplemented from gdcm::DataElement.

References gdcm::DataSet::Clear(), gdcmDebugMacro, gdcmErrorMacro, gdcmWarningMacro, gdcm::DataSet::Is-Empty(), and gdcm::SwapperDoOp::Swap().

Referenced by gdcm::SequenceOfItems::Read().

25.148.3.9 void gdcm::Item::SetNestedDataSet (const DataSet & *nested*) [inline]

25.148.3.10 template<typename TDE , typename TSwap > const std::ostream& gdcm::Item::Write (std::ostream & *os*) const [inline]

Reimplemented from gdcm::DataElement.

References gdcmWarningMacro, gdcm::VL::GetLength(), gdcm::VL::Write(), and gdcm::Tag::Write().

25.148.4 Friends And Related Function Documentation

25.148.4.1 std::ostream& operator<< (std::ostream & *os,* const Item & *val*) [friend]

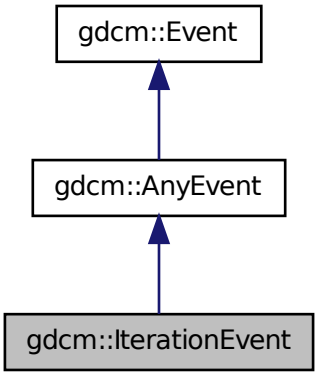
The documentation for this class was generated from the following file:

- gdcmItem.h

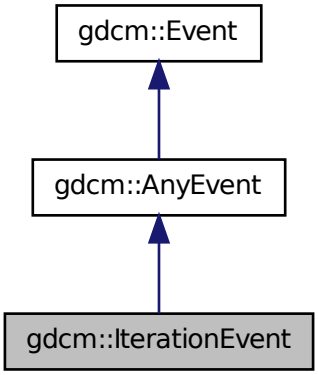
25.149 gdcm::IterationEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::IterationEvent:



Collaboration diagram for gdcm::IterationEvent:



The documentation for this class was generated from the following file:

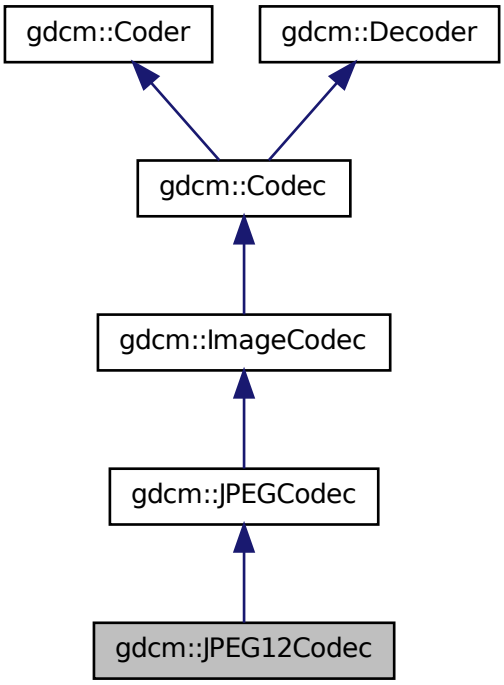
- gdcmEvent.h

25.150 gdcm::JPEG12Codec Class Reference

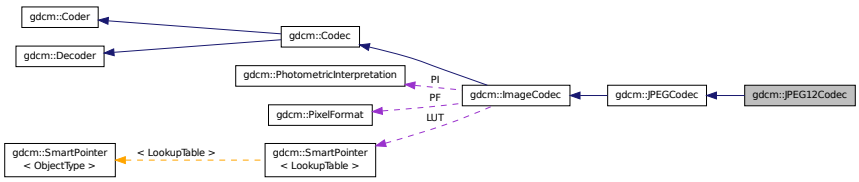
Class to do JPEG 12bits (lossy & lossless)

```
#include <gdcmJPEG12Codec.h>
```

Inheritance diagram for gdcm::JPEG12Codec:



Collaboration diagram for gdcm::JPEG12Codec:



Public Member Functions

- JPEG12Codec ()
- ∼JPEG12Codec ()
- bool Decode (std::istream &is, std::ostream &os)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool InternalCode (const char ∗input, unsigned long len, std::ostream &os)

Additional Inherited Members

25.150.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

Note

internal class

25.150.2 Constructor & Destructor Documentation

25.150.2.1 gdcm::JPEG12Codec::JPEG12Codec ( )

25.150.2.2 gdcm::JPEG12Codec::~JPEG12Codec ( )

25.150.3 Member Function Documentation

25.150.3.1 bool gdcm::JPEG12Codec::Decode (std::istream & *is*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::JPEGCodec.

25.150.3.2 bool gdcm::JPEG12Codec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::JPEGCodec.

25.150.3.3 bool gdcm::JPEG12Codec::InternalCode (const char * *input*, unsigned long *len*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::Coder.

The documentation for this class was generated from the following file:

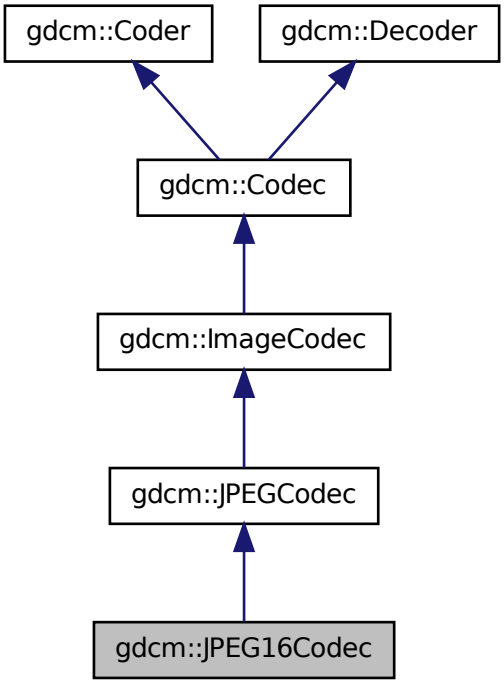
- gdcmJPEG12Codec.h

25.151 gdcm::JPEG16Codec Class Reference

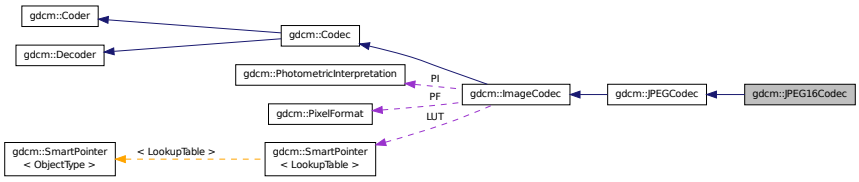
Class to do JPEG 16bits (lossless)

```
#include <gdcmJPEG16Codec.h>
```

Inheritance diagram for gdcm::JPEG16Codec:



Collaboration diagram for gdcm::JPEG16Codec:



Public Member Functions

- JPEG16Codec ()
- ~JPEG16Codec ()
- bool Decode (std::istream &is, std::ostream &os)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool InternalCode (const char *input, unsigned long len, std::ostream &os)

Additional Inherited Members

25.151.1 Detailed Description

Class to do JPEG 16bits (lossless)

Note

internal class

25.151.2 Constructor & Destructor Documentation

25.151.2.1 gdcm::JPEG16Codec::JPEG16Codec ( )

25.151.2.2 gdcm::JPEG16Codec::~JPEG16Codec ( )

25.151.3 Member Function Documentation

25.151.3.1 bool gdcm::JPEG16Codec::Decode (std::istream & *is*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::JPEGCodec.

25.151.3.2 bool gdcm::JPEG16Codec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::JPEGCodec.

25.151.3.3 bool gdcm::JPEG16Codec::InternalCode (const char * *input*, unsigned long *len*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::Coder.

The documentation for this class was generated from the following file:

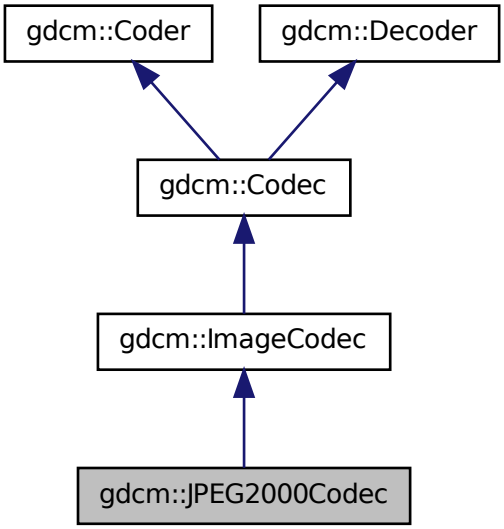
- gdcmJPEG16Codec.h

25.152 gdcm::JPEG2000Codec Class Reference

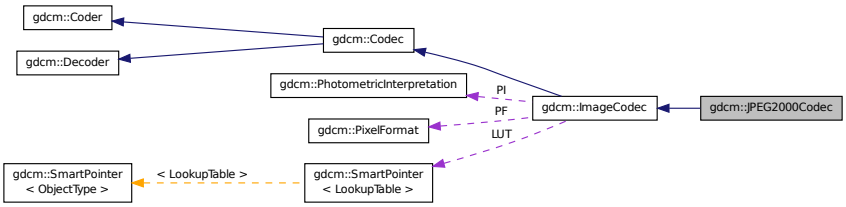
Class to do JPEG 2000.

```
#include <gdcmJPEG2000Codec.h>
```

Inheritance diagram for gdcm::JPEG2000Codec:



Collaboration diagram for gdcm::JPEG2000Codec:



Public Member Functions

- JPEG2000Codec ()
- ~JPEG2000Codec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

- virtual bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- double GetQuality (unsigned int idx=0) const
- double GetRate (unsigned int idx=0) const
- void SetNumberOfResolutions (unsigned int nres)
- void SetQuality (unsigned int idx, double q)
- void SetRate (unsigned int idx, double rate)
- void SetReversible (bool res)
- void SetTileSize (unsigned int tx, unsigned int ty)

Protected Member Functions

- bool Decode (std::istream &is, std::ostream &os)

Friends

- class Bitmap

Additional Inherited Members

25.152.1 Detailed Description

Class to do JPEG 2000.

Note

the class will produce JPC (JPEG 2000 codestream), since some private implementor are using full jp2 file the decoder tolerate jp2 input this is an implementation of an ImageCodec

25.152.2 Constructor & Destructor Documentation

25.152.2.1 gdcm::JPEG2000Codec::JPEG2000Codec ( )

25.152.2.2 gdcm::JPEG2000Codec::~JPEG2000Codec ( )

25.152.3 Member Function Documentation

25.152.3.1 bool gdcm::JPEG2000Codec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.152.3.2 bool gdcm::JPEG2000Codec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.152.3.3 bool gdcm::JPEG2000Codec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

25.152.3.4 bool gdcm::JPEG2000Codec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

25.152.3.5 bool gdcm::JPEG2000Codec::Decode (std::istream & *is*, std::ostream & *os*) [protected],[virtual]

Reimplemented from gdcm::ImageCodec.

25.152.3.6 virtual bool gdcm::JPEG2000Codec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

25.152.3.7 double gdcm::JPEG2000Codec::GetQuality (unsigned int *idx* = 0) const

25.152.3.8 double gdcm::JPEG2000Codec::GetRate (unsigned int *idx* = 0) const

25.152.3.9 void gdcm::JPEG2000Codec::SetNumberOfResolutions (unsigned int *nres*)

25.152.3.10 void gdcm::JPEG2000Codec::SetQuality (unsigned int *idx*, double *q*)

25.152.3.11 void gdcm::JPEG2000Codec::SetRate (unsigned int *idx*, double *rate*)

25.152.3.12 void gdcm::JPEG2000Codec::SetReversible (bool *res*)

25.152.3.13 void gdcm::JPEG2000Codec::SetTileSize (unsigned int *tx*, unsigned int *ty*)

25.152.4 Friends And Related Function Documentation

25.152.4.1 friend class Bitmap [friend]

The documentation for this class was generated from the following file:

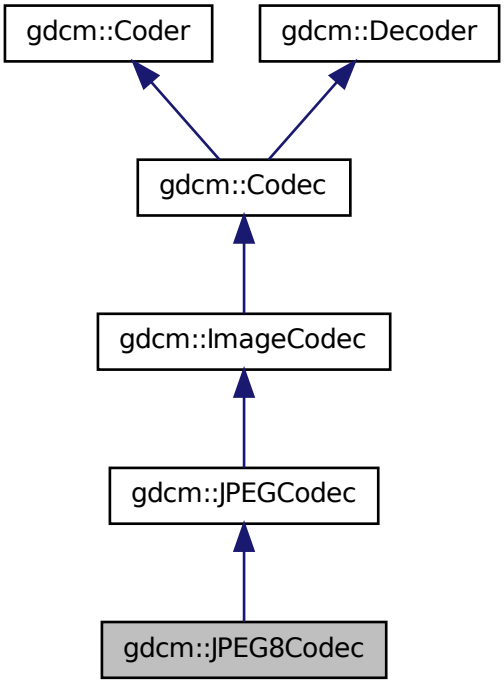
- gdcmJPEG2000Codec.h

25.153 gdcm::JPEG8Codec Class Reference

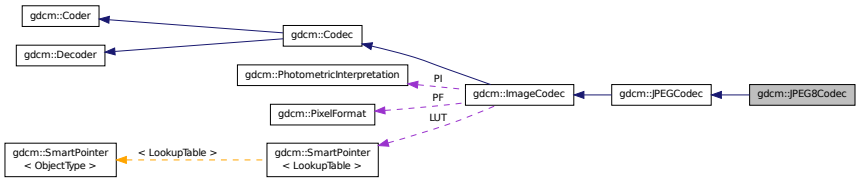
Class to do JPEG 8bits (lossy & lossless)

```
#include <gdcmJPEG8Codec.h>
```

Inheritance diagram for gdcm::JPEG8Codec:



Collaboration diagram for gdcm::JPEG8Codec:



Public Member Functions

- JPEG8Codec ()
- ∼JPEG8Codec ()
- bool Decode (std::istream &is, std::ostream &os)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool InternalCode (const char ∗input, unsigned long len, std::ostream &os)

Additional Inherited Members

25.153.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

Note

internal class

25.153.2 Constructor & Destructor Documentation

25.153.2.1 gdcm::JPEG8Codec::JPEG8Codec ( )

25.153.2.2 gdcm::JPEG8Codec::∼JPEG8Codec ( )

25.153.3 Member Function Documentation

25.153.3.1 bool gdcm::JPEG8Codec::Decode (std::istream & *is,* std::ostream & *os*) [virtual]

Reimplemented from gdcm::JPEGCodec.

25.153.3.2 bool gdcm::JPEG8Codec::GetHeaderInfo (std::istream & *is,* TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::JPEGCodec.

25.153.3.3 bool gdcm::JPEG8Codec::InternalCode (const char ∗ *input,* unsigned long *len,* std::ostream & *os*) [virtual]

Reimplemented from gdcm::Coder.

The documentation for this class was generated from the following file:

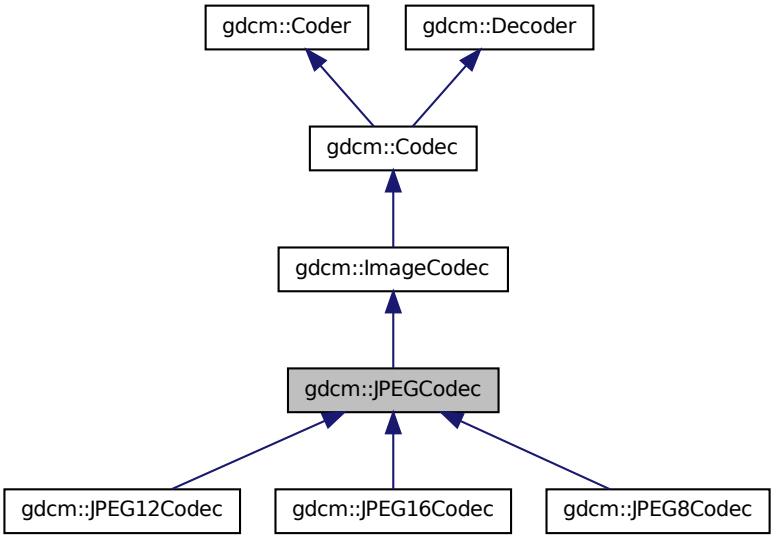
- gdcmJPEG8Codec.h

25.154 gdcm::JPEGCodec Class Reference

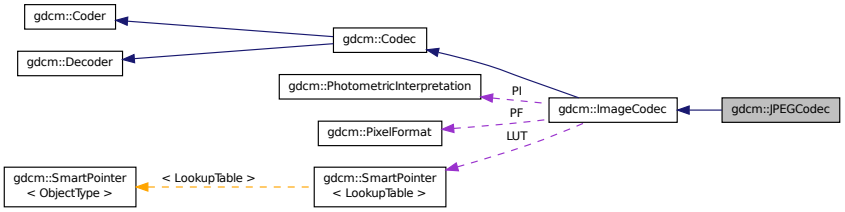
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: gdcm::JPEG8Codec, gdcm::JPEG12Codec & gdcm::JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case.

```
#include <gdcmJPEGCodec.h>
```

Inheritance diagram for gdcm::JPEGCodec:



Collaboration diagram for gdcm::JPEGCodec:



Public Member Functions

- JPEGCodec ()
- ~JPEGCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Code (DataElement const &in, DataElement &out)

  Compress into JPEG.
- void ComputeOffsetTable (bool b)

  Compute the offset table:

- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- virtual bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool GetLossless () const
- double GetQuality () const
- void SetLossless (bool l)
- void SetPixelFormat (PixelFormat const &pf)
- void SetQuality (double q)

Protected Member Functions

- bool Decode (std::istream &is, std::ostream &os)
- bool IsValid (PhotometricInterpretation const &pi)
- void SetBitSample (int bit)

Protected Attributes

- int BitSample
- bool Lossless
- int Quality

Additional Inherited Members

25.154.1 Detailed Description

JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: gdcm::JPEG8Codec, gdcm::JPEG12Codec & gdcm::JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case.

Note

Things you should know if you ever want to dive into DICOM/JPEG world (among other):

- `http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/625e46919f208`
- `http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/75fdfccc65a62`
- `http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f09`
- `http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c9`

Examples:

GetJPEGSamplePrecision.cxx.

25.154.2 Constructor & Destructor Documentation

25.154.2.1 gdcm::JPEGCodec::JPEGCodec ( )

25.154.2.2 gdcm::JPEGCodec::∼JPEGCodec ( )

25.154.3 Member Function Documentation

25.154.3.1 bool gdcm::JPEGCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.154.3.2 bool gdcm::JPEGCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.154.3.3 bool gdcm::JPEGCodec::Code (DataElement const & *in*, DataElement & *out*) [virtual]

Compress into JPEG.

Reimplemented from gdcm::Coder.

25.154.3.4 void gdcm::JPEGCodec::ComputeOffsetTable (bool *b*)

Compute the offset table:

25.154.3.5 bool gdcm::JPEGCodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

25.154.3.6 bool gdcm::JPEGCodec::Decode (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

25.154.3.7 virtual bool gdcm::JPEGCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

Examples:

GetJPEGSamplePrecision.cxx.

25.154.3.8 bool gdcm::JPEGCodec::GetLossless ( ) const

25.154.3.9 double gdcm::JPEGCodec::GetQuality ( ) const

25.154.3.10 bool gdcm::JPEGCodec::IsValid (PhotometricInterpretation const & *pi*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

25.154.3.11 void gdcm::JPEGCodec::SetBitSample (int *bit*) [protected]

25.154.3.12 void gdcm::JPEGCodec::SetLossless (bool *l*)

25.154.3.13 void gdcm::JPEGCodec::SetPixelFormat (PixelFormat const & *pf*) [virtual]

Reimplemented from gdcm::ImageCodec.

Examples:

GetJPEGSamplePrecision.cxx.

25.154.3.14 void gdcm::JPEGCodec::SetQuality (double *q*)

25.154.4 Member Data Documentation

25.154.4.1 int gdcm::JPEGCodec::BitSample [protected]

25.154.4.2 bool gdcm::JPEGCodec::Lossless [protected]

25.154.4.3 int gdcm::JPEGCodec::Quality [protected]

The documentation for this class was generated from the following file:

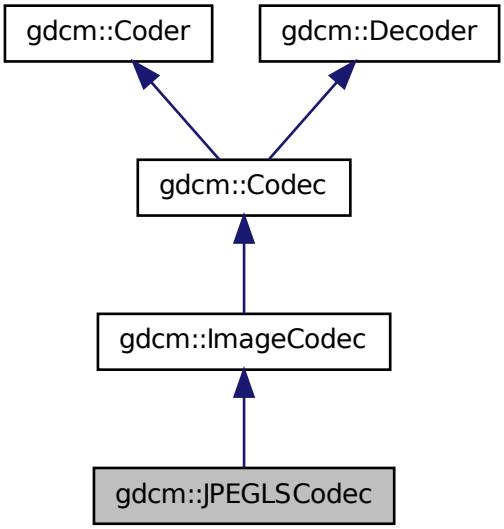
- gdcmJPEGCodec.h

25.155 gdcm::JPEGLSCodec Class Reference

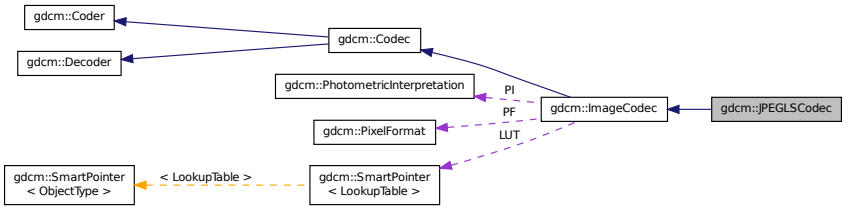
JPEG-LS.

```
#include <gdcmJPEGLSCodec.h>
```

Inheritance diagram for gdcm::JPEGLSCodec:



Collaboration diagram for gdcm::JPEGLSCodec:



Public Member Functions

- JPEGLSCodec ()
- ~JPEGLSCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

Decode.

- bool Decode (DataElement const &in, char ∗outBuffer, uint32_t inBufferLength, uint32_t inXMin, uint32_t inXMax, uint32_t inYMin, uint32_t inYMax, uint32_t inZMin, uint32_t inZMax)
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool GetLossless () const
- void SetBufferLength (unsigned long l)
- void SetLossless (bool l)
- void SetLossyError (int error)

  [0-3] generally

Additional Inherited Members

25.155.1 Detailed Description

JPEG-LS.

Note

codec that implement the JPEG-LS compression this is an implementation of ImageCodec for JPEG-LS

It uses the CharLS JPEG-LS implementation `http://charls.codeplex.com`

25.155.2 Constructor & Destructor Documentation

25.155.2.1 gdcm::JPEGLSCodec::JPEGLSCodec ( )

25.155.2.2 gdcm::JPEGLSCodec::∼JPEGLSCodec ( )

25.155.3 Member Function Documentation

25.155.3.1 bool gdcm::JPEGLSCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.155.3.2 bool gdcm::JPEGLSCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.155.3.3 bool gdcm::JPEGLSCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

25.155.3.4 bool gdcm::JPEGLSCodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

25.155.3.5 bool gdcm::JPEGLSCodec::Decode (DataElement const & *in*, char * *outBuffer*, uint32_t *inBufferLength*, uint32_t *inXMin*, uint32_t *inXMax*, uint32_t *inYMin*, uint32_t *inYMax*, uint32_t *inZMin*, uint32_t *inZMax*)

25.155.3.6 unsigned long gdcm::JPEGLSCodec::GetBufferLength ( ) const [inline]

25.155.3.7 bool gdcm::JPEGLSCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

25.155.3.8 bool gdcm::JPEGLSCodec::GetLossless ( ) const

25.155.3.9 void gdcm::JPEGLSCodec::SetBufferLength (unsigned long *l*) [inline]

25.155.3.10 void gdcm::JPEGLSCodec::SetLossless (bool *l*)

25.155.3.11 void gdcm::JPEGLSCodec::SetLossyError (int *error*)

[0-3] generally

The documentation for this class was generated from the following file:

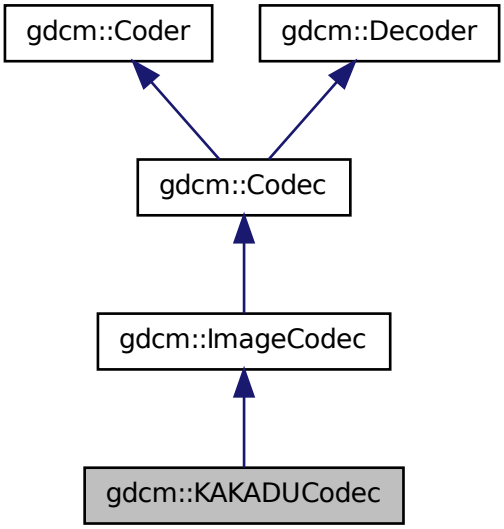
- gdcmJPEGLSCodec.h

25.156 gdcm::KAKADUCodec Class Reference

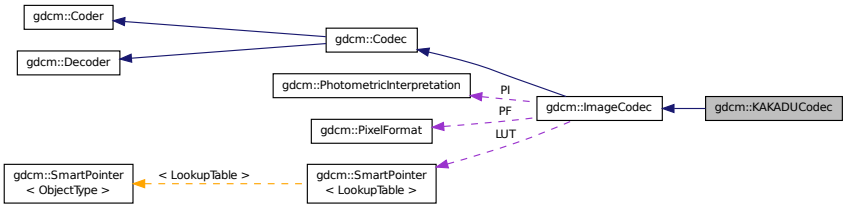
KAKADUCodec.

```
#include <gdcmKAKADUCodec.h>
```

Inheritance diagram for gdcm::KAKADUCodec:



Collaboration diagram for gdcm::KAKADUCodec:



Public Member Functions

- KAKADUCodec ()
- ~KAKADUCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Additional Inherited Members

25.156.1 Detailed Description

KAKADUCodec.

25.156.2 Constructor & Destructor Documentation

25.156.2.1 gdcm::KAKADUCodec::KAKADUCodec ( )

25.156.2.2 gdcm::KAKADUCodec::~KAKADUCodec ( )

25.156.3 Member Function Documentation

25.156.3.1 bool gdcm::KAKADUCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.156.3.2 bool gdcm::KAKADUCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.156.3.3 bool gdcm::KAKADUCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

25.156.3.4 bool gdcm::KAKADUCodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

The documentation for this class was generated from the following file:

- gdcmKAKADUCodec.h

25.157 gdcm::LO Class Reference

LO.

```
#include <gdcmLO.h>
```

Public Types

- typedef Superclass::const_iterator const_iterator

- typedef Superclass::const_reference const_reference
- typedef Superclass::const_reverse_iterator const_reverse_iterator
- typedef Superclass::difference_type difference_type
- typedef Superclass::iterator iterator
- typedef Superclass::pointer pointer
- typedef Superclass::reference reference
- typedef Superclass::reverse_iterator reverse_iterator
- typedef Superclass::size_type size_type
- typedef String<'\\', 64 > Superclass
- typedef Superclass::value_type value_type

Public Member Functions

- LO ()
- LO (const value_type *s)
- LO (const value_type *s, size_type n)
- LO (const Superclass &s, size_type pos=0, size_type n=npos)
- bool IsValid () const

25.157.1 Detailed Description

LO.

Note

TODO

25.157.2 Member Typedef Documentation

25.157.2.1 typedef Superclass::const_iterator gdcm::LO::const_iterator

25.157.2.2 typedef Superclass::const_reference gdcm::LO::const_reference

25.157.2.3 typedef Superclass::const_reverse_iterator gdcm::LO::const_reverse_iterator

25.157.2.4 typedef Superclass::difference_type gdcm::LO::difference_type

25.157.2.5 typedef Superclass::iterator gdcm::LO::iterator

25.157.2.6 typedef Superclass::pointer gdcm::LO::pointer

25.157.2.7 typedef Superclass::reference gdcm::LO::reference

25.157.2.8 typedef Superclass::reverse_iterator gdcm::LO::reverse_iterator

25.157.2.9 typedef Superclass::size_type gdcm::LO::size_type

25.157.2.10 typedef String<'\\',64> gdcm::LO::Superclass

25.157.2.11 typedef Superclass::value_type gdcm::LO::value_type

25.157.3 Constructor & Destructor Documentation

25.157.3.1 gdcm::LO::LO ( ) [inline]

25.157.3.2 gdcm::LO::LO (const value_type * *s*) [inline]

25.157.3.3 gdcm::LO::LO (const value_type * *s*, size_type *n*) [inline]

25.157.3.4 gdcm::LO::LO (const Superclass & *s*, size_type *pos* = 0, size_type *n* = npos) [inline]

25.157.4 Member Function Documentation

25.157.4.1 bool gdcm::LO::IsValid ( ) const [inline]

References gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid().

The documentation for this class was generated from the following file:

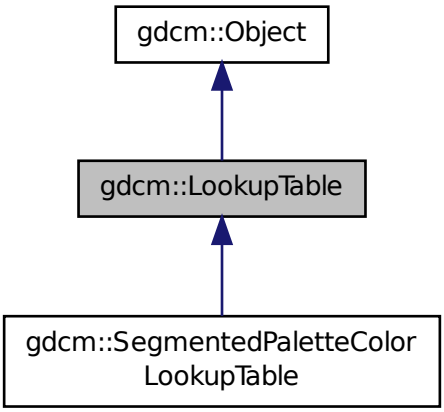
- gdcmLO.h

25.158 gdcm::LookupTable Class Reference

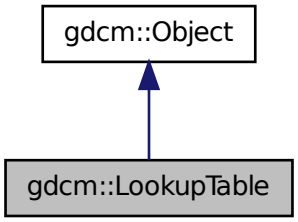
LookupTable class.

`#include <gdcmLookupTable.h>`

Inheritance diagram for gdcm::LookupTable:



Collaboration diagram for gdcm::LookupTable:



Public Types

- enum LookupTableType {
  RED = 0,
  GREEN,
  BLUE,
  GRAY,
  UNKNOWN }

Public Member Functions

- LookupTable ()
- LookupTable (LookupTable const &lut)
- ~LookupTable ()
- void Allocate (unsigned short bitsample=8)

  Allocate the LUT.
- void Clear ()

  Clear the LUT.
- void Decode (std::istream &is, std::ostream &os) const

  Decode the LUT.
- unsigned short GetBitSample () const

  return the bit sample
- bool GetBufferAsRGBA (unsigned char *rgba) const

  return the LUT as RGBA buffer
- void GetLUT (LookupTableType type, unsigned char *array, unsigned int &length) const
- void GetLUTDescriptor (LookupTableType type, unsigned short &length, unsigned short &subscript, unsigned short &bitsize) const
- unsigned int GetLUTLength (LookupTableType type) const
- const unsigned char * GetPointer () const

  return a raw pointer to the LUT
- void InitializeBlueLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)
- bool Initialized () const

  return whether the LUT has been initialized

- void InitializeGreenLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)
- void InitializeLUT (LookupTableType type, unsigned short length, unsigned short subscript, unsigned short bitsize)

  Generic interface:
- void InitializeRedLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)

  RED / GREEN / BLUE specific:
- void Print (std::ostream &) const
- void SetBlueLUT (const unsigned char ∗blue, unsigned int length)
- void SetGreenLUT (const unsigned char ∗green, unsigned int length)
- virtual void SetLUT (LookupTableType type, const unsigned char ∗array, unsigned int length)
- void SetRedLUT (const unsigned char ∗red, unsigned int length)
- bool WriteBufferAsRGBA (const unsigned char ∗rgba)

  Write the LUT as RGBA.

Protected Attributes

- unsigned short BitSample
- bool IncompleteLUT:1
- LookupTableInternal ∗ Internal

Additional Inherited Members

25.158.1 Detailed Description

LookupTable class.

25.158.2 Member Enumeration Documentation

25.158.2.1 enum gdcm::LookupTable::LookupTableType

Enumerator:

RED

GREEN

BLUE

GRAY

UNKNOWN

25.158.3 Constructor & Destructor Documentation

25.158.3.1 gdcm::LookupTable::LookupTable ( )

25.158.3.2 gdcm::LookupTable::∼LookupTable ( )

25.158.3.3 gdcm::LookupTable::LookupTable (LookupTable const & *lut*) [inline]

25.158.4 Member Function Documentation

25.158.4.1 void gdcm::LookupTable::Allocate (unsigned short *bitsample* = 8)

Allocate the LUT.

25.158.4.2 void gdcm::LookupTable::Clear ( )

Clear the LUT.

25.158.4.3 void gdcm::LookupTable::Decode (std::istream & *is*, std::ostream & *os*) const

Decode the LUT.

25.158.4.4 unsigned short gdcm::LookupTable::GetBitSample ( ) const [inline]

return the bit sample

25.158.4.5 bool gdcm::LookupTable::GetBufferAsRGBA (unsigned char * *rgba*) const

return the LUT as RGBA buffer

25.158.4.6 void gdcm::LookupTable::GetLUT (LookupTableType *type*, unsigned char * *array*, unsigned int & *length*) const

25.158.4.7 void gdcm::LookupTable::GetLUTDescriptor (LookupTableType *type*, unsigned short & *length*, unsigned short & *subscript*, unsigned short & *bitsize*) const

25.158.4.8 unsigned int gdcm::LookupTable::GetLUTLength (LookupTableType *type*) const

25.158.4.9 const unsigned char* gdcm::LookupTable::GetPointer ( ) const

return a raw pointer to the LUT

25.158.4.10 void gdcm::LookupTable::InitializeBlueLUT (unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize*)

25.158.4.11 bool gdcm::LookupTable::Initialized ( ) const

return whether the LUT has been initialized

25.158.4.12 void gdcm::LookupTable::InitializeGreenLUT (unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize*)

25.158.4.13 void gdcm::LookupTable::InitializeLUT (LookupTableType *type*, unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize*)

Generic interface:

25.158.4.14 void gdcm::LookupTable::InitializeRedLUT (unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize*)

RED / GREEN / BLUE specific:

25.158.4.15 void gdcm::LookupTable::Print (std::ostream &) const [inline], [virtual]

Reimplemented from gdcm::Object.

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

25.158.4.16 void gdcm::LookupTable::SetBlueLUT (const unsigned char * *blue,* unsigned int *length*)

25.158.4.17 void gdcm::LookupTable::SetGreenLUT (const unsigned char * *green,* unsigned int *length*)

25.158.4.18 virtual void gdcm::LookupTable::SetLUT (LookupTableType *type,* const unsigned char * *array,* unsigned int *length*) [virtual]

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

25.158.4.19 void gdcm::LookupTable::SetRedLUT (const unsigned char * *red,* unsigned int *length*)

25.158.4.20 bool gdcm::LookupTable::WriteBufferAsRGBA (const unsigned char * *rgba*)

Write the LUT as RGBA.

25.158.5 Member Data Documentation

25.158.5.1 unsigned short gdcm::LookupTable::BitSample [protected]

25.158.5.2 bool gdcm::LookupTable::IncompleteLUT [protected]

25.158.5.3 LookupTableInternal* gdcm::LookupTable::Internal [protected]

The documentation for this class was generated from the following file:

- gdcmLookupTable.h

25.159 gdcm::Scanner::ltstr Struct Reference

```
#include <gdcmScanner.h>
```

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

25.159.1 Member Function Documentation

25.159.1.1 bool gdcm::Scanner::ltstr::operator() (const char * *s1,* const char * *s2*) const [inline]

The documentation for this struct was generated from the following file:

- gdcmScanner.h

25.160 gdcm::Macro Class Reference

Class for representing a Macro.

`#include <gdcmMacro.h>`

Public Types

- typedef std::vector< std::string > ArrayIncludeMacrosType
- typedef std::map< Tag, MacroEntry > MapModuleEntry

Public Member Functions

- Macro ()
- void AddMacroEntry (const Tag &tag, const MacroEntry &module)

  Will add a ModuleEntry direcly at root-level. See Macro for nested-included level.
- void Clear ()
- bool FindMacroEntry (const Tag &tag) const
- const MacroEntry & GetMacroEntry (const Tag &tag) const
- const char * GetName () const
- void SetName (const char *name)
- bool Verify (const DataSet &ds, Usage const &usage) const

Friends

- std::ostream & operator<< (std::ostream &_os, const Macro &_val)

25.160.1 Detailed Description

Class for representing a Macro.

Note

Attribute Macro: a set of Attributes that are described in a single table that is referenced by multiple Module or other tables.

See also

Module

25.160.2 Member Typedef Documentation

25.160.2.1 typedef std::vector<std::string> gdcm::Macro::ArrayIncludeMacrosType

25.160.2.2 typedef std::map<Tag, MacroEntry> gdcm::Macro::MapModuleEntry

25.160.3 Constructor & Destructor Documentation

25.160.3.1 gdcm::Macro::Macro ( ) [inline]

25.160.4 Member Function Documentation

25.160.4.1 void gdcm::Macro::AddMacroEntry (const Tag & *tag,* const MacroEntry & *module*) [inline]

Will add a ModuleEntry direcly at root-level. See Macro for nested-included level.

25.160.4.2 void gdcm::Macro::Clear ( ) [inline]

25.160.4.3 bool gdcm::Macro::FindMacroEntry (const Tag & *tag*) const

Find or Get a ModuleEntry. ModuleEntry are either search are root-level or within nested-macro included in module.

25.160.4.4 const MacroEntry& gdcm::Macro::GetMacroEntry (const Tag & *tag*) const

25.160.4.5 const char* gdcm::Macro::GetName ( ) const [inline]

25.160.4.6 void gdcm::Macro::SetName (const char * *name*) [inline]

25.160.4.7 bool gdcm::Macro::Verify (const DataSet & *ds,* Usage const & *usage*) const

25.160.5 Friends And Related Function Documentation

25.160.5.1 std::ostream& operator<< (std::ostream & *_os,* const Macro & *_val*) [friend]

The documentation for this class was generated from the following file:

- gdcmMacro.h

25.161 gdcm::Macros Class Reference

Class for representing a Modules.

`#include <gdcmMacros.h>`

Public Types

- typedef std::map< std::string,
  Macro > ModuleMapType

Public Member Functions

- Macros ()
- void AddMacro (const char *ref, const Macro &module)
- void Clear ()
- const Macro & GetMacro (const char *name) const
- bool IsEmpty () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Macros &_val)

25.161.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples:

TraverseModules.cxx.

25.161.2 Member Typedef Documentation

25.161.2.1 typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType

25.161.3 Constructor & Destructor Documentation

25.161.3.1 gdcm::Macros::Macros ( ) [inline]

25.161.4 Member Function Documentation

25.161.4.1 void gdcm::Macros::AddMacro (const char * *ref*, const Macro & *module*) [inline]

25.161.4.2 void gdcm::Macros::Clear ( ) [inline]

25.161.4.3 const Macro& gdcm::Macros::GetMacro (const char * *name*) const [inline]

25.161.4.4 bool gdcm::Macros::IsEmpty ( ) const [inline]

25.161.5 Friends And Related Function Documentation

25.161.5.1 std::ostream& operator<< (std::ostream & *_os*, const Macros & *_val*) [friend]

The documentation for this class was generated from the following file:

- gdcmMacros.h

25.162 gdcm::network::MaximumLengthSub Class Reference

MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#include <gdcmMaximumLengthSub.h>
```

Public Member Functions

- MaximumLengthSub ()
- uint32_t GetMaximumLength () const
- std::istream & Read (std::istream &is)
- void SetMaximumLength (uint32_t maximumlength)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.162.1 Detailed Description

MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

or

Table D.1-2 Maximum length sub-item fields (A-ASSOCIATE-AC)

25.162.2 Constructor & Destructor Documentation

25.162.2.1 gdcm::network::MaximumLengthSub::MaximumLengthSub ( )

25.162.3 Member Function Documentation

25.162.3.1 uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength ( ) const [inline]

25.162.3.2 std::istream& gdcm::network::MaximumLengthSub::Read (std::istream & *is*)

25.162.3.3 void gdcm::network::MaximumLengthSub::SetMaximumLength (uint32_t *maximumlength*) [inline]

25.162.3.4 size_t gdcm::network::MaximumLengthSub::Size ( ) const

25.162.3.5 const std::ostream& gdcm::network::MaximumLengthSub::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

- gdcmMaximumLengthSub.h

25.163 gdcm::MD5 Class Reference

Class for MD5.

```
#include <gdcmMD5.h>
```

Public Member Functions

- MD5 ()
- ~MD5 ()

Static Public Member Functions

- static bool Compute (const char ∗buffer, unsigned long buf_len, char digest_str[33])
- static bool ComputeFile (const char ∗filename, char digest_str[33])

25.163.1 Detailed Description

Class for MD5.

Warning

this class is able to pick from two implementations:

1. a lightweight md5 implementation (when GDCM_BUILD_TESTING is turned ON)
2. the one from OpenSSL (when GDCM_USE_SYSTEM_OPENSSL is turned ON)

In all other cases it will return an error

25.163.2 Constructor & Destructor Documentation

25.163.2.1 gdcm::MD5::MD5 ( )

25.163.2.2 gdcm::MD5::∼MD5 ( )

25.163.3 Member Function Documentation

25.163.3.1 static bool gdcm::MD5::Compute (const char ∗ *buffer,* unsigned long *buf_len,* char *digest_str[33]*) [static]

25.163.3.2 static bool gdcm::MD5::ComputeFile (const char ∗ *filename,* char *digest_str[33]*) [static]

The documentation for this class was generated from the following file:

- gdcmMD5.h

25.164 gdcm::MediaStorage Class Reference

MediaStorage.

```
#include <gdcmMediaStorage.h>
```

Public Types

- enum MSType {
  MediaStorageDirectoryStorage = 0,
  ComputedRadiographyImageStorage,
  DigitalXRayImageStorageForPresentation,
  DigitalXRayImageStorageForProcessing,
  DigitalMammographyImageStorageForPresentation,
  DigitalMammographyImageStorageForProcessing,
  DigitalIntraoralXrayImageStorageForPresentation,
  DigitalIntraoralXRayImageStorageForProcessing,
  CTImageStorage,
  EnhancedCTImageStorage,
  UltrasoundImageStorageRetired,
  UltrasoundImageStorage,
  UltrasoundMultiFrameImageStorageRetired,
  UltrasoundMultiFrameImageStorage,
  MRImageStorage,
  EnhancedMRImageStorage,
  MRSpectroscopyStorage,
  NuclearMedicineImageStorageRetired,
  SecondaryCaptureImageStorage,
  MultiframeSingleBitSecondaryCaptureImageStorage,
  MultiframeGrayscaleByteSecondaryCaptureImageStorage,
  MultiframeGrayscaleWordSecondaryCaptureImageStorage,
  MultiframeTrueColorSecondaryCaptureImageStorage,
  StandaloneOverlayStorage,
  StandaloneCurveStorage,
  LeadECGWaveformStorage,
  GeneralECGWaveformStorage,
  AmbulatoryECGWaveformStorage,
  HemodynamicWaveformStorage,
  CardiacElectrophysiologyWaveformStorage,
  BasicVoiceAudioWaveformStorage,
  StandaloneModalityLUTStorage,
  StandaloneVOILUTStorage,
  GrayscaleSoftcopyPresentationStateStorageSOPClass,
  XRayAngiographicImageStorage,
  XRayRadiofluoroscopingImageStorage,
  XRayAngiographicBiPlaneImageStorageRetired,
  NuclearMedicineImageStorage,
  RawDataStorage,
  SpacialRegistrationStorage,
  SpacialFiducialsStorage,
  PETImageStorage,
  RTImageStorage,
  RTDoseStorage,
  RTStructureSetStorage,
  RTPlanStorage,
  CSANonImageStorage,
  Philips3D,
  EnhancedSR,
  BasicTextSR,
  HardcopyGrayscaleImageStorage,
  ComprehensiveSR,
  DetachedStudyManagementSOPClass,
  EncapsulatedPDFStorage,
  StudyComponentManagementSOPClass,
  DetachedVisitManagementSOPClass,
  DetachedPatientManagementSOPClass,
  VideoEndoscopicImageStorage,

MS_END }

- enum ObjectType {
  NoObject = 0,
  Video,
  Waveform,
  Audio,
  PDF,
  URI,
  Segmentation,
  ObjectEnd }

Public Member Functions

- MediaStorage (MSType type=MS_END)
- const char ∗ GetModality () const
- unsigned int GetModalityDimension () const
- const char ∗ GetString () const

  Return the Media String of the object.
- void GuessFromModality (const char ∗modality, unsigned int dimension=2)
- bool IsUndefined () const
- operator MSType () const
- bool SetFromDataSet (DataSet const &ds)
- bool SetFromFile (File const &file)
- bool SetFromHeader (FileMetaInformation const &fmi)
- bool SetFromModality (DataSet const &ds)

Static Public Member Functions

- static const char ∗ GetMSString (MSType ts)

  Return the Media String associated. Will return NULL for MS_END.
- static MSType GetMSType (const char ∗str)
- static unsigned int GetNumberOfModality ()
- static unsigned int GetNumberOfMSString ()
- static unsigned int GetNumberOfMSType ()
- static bool IsImage (MSType ts)

Protected Member Functions

- void SetFromSourceImageSequence (DataSet const &ds)

Friends

- std::ostream & operator<< (std::ostream &os, const MediaStorage &ms)

25.164.1 Detailed Description

MediaStorage.

Note

FIXME There should not be any notion of Image and/or PDF at that point Only the codec can answer yes I support this Media Storage or not... For instance an ImageCodec will answer yes to most of them while a PDFCodec will answer only for the Encapsulated PDF

See also

UIDs

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_-Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, ReadAndDumpDICOMDIR.cxx, Stream-ImageReaderTest.cxx, and TestReader.cxx.

25.164.2 Member Enumeration Documentation

25.164.2.1 enum gdcm::MediaStorage::MSType

Enumerator:

MediaStorageDirectoryStorage

ComputedRadiographyImageStorage

DigitalXRayImageStorageForPresentation

DigitalXRayImageStorageForProcessing

DigitalMammographyImageStorageForPresentation

DigitalMammographyImageStorageForProcessing

DigitalIntraoralXrayImageStorageForPresentation

DigitalIntraoralXRayImageStorageForProcessing

CTImageStorage

EnhancedCTImageStorage

UltrasoundImageStorageRetired

UltrasoundImageStorage

UltrasoundMultiFrameImageStorageRetired

UltrasoundMultiFrameImageStorage

MRImageStorage

EnhancedMRImageStorage

MRSpectroscopyStorage

NuclearMedicineImageStorageRetired

SecondaryCaptureImageStorage

MultiframeSingleBitSecondaryCaptureImageStorage

MultiframeGrayscaleByteSecondaryCaptureImageStorage

MultiframeGrayscaleWordSecondaryCaptureImageStorage

MultiframeTrueColorSecondaryCaptureImageStorage

StandaloneOverlayStorage

StandaloneCurveStorage

LeadECGWaveformStorage

GeneralECGWaveformStorage

AmbulatoryECGWaveformStorage

HemodynamicWaveformStorage

CardiacElectrophysiologyWaveformStorage

BasicVoiceAudioWaveformStorage

StandaloneModalityLUTStorage

StandaloneVOILUTStorage

GrayscaleSoftcopyPresentationStateStorageSOPClass

XRayAngiographicImageStorage

XRayRadiofluoroscopingImageStorage

XRayAngiographicBiPlaneImageStorageRetired

NuclearMedicineImageStorage

RawDataStorage

SpacialRegistrationStorage

SpacialFiducialsStorage

PETImageStorage

RTImageStorage

RTDoseStorage

RTStructureSetStorage

RTPlanStorage

CSANonImageStorage

Philips3D

EnhancedSR

BasicTextSR

HardcopyGrayscaleImageStorage

ComprehensiveSR

DetachedStudyManagementSOPClass

EncapsulatedPDFStorage

StudyComponentManagementSOPClass

DetachedVisitManagementSOPClass

DetachedPatientManagementSOPClass

VideoEndoscopicImageStorage

GeneralElectricMagneticResonanceImageStorage

GEPrivate3DModelStorage

ToshibaPrivateDataStorage

MammographyCADSR

KeyObjectSelectionDocument

HangingProtocolStorage
ModalityPerformedProcedureStepSOPClass
PhilipsPrivateMRSyntheticImageStorage
VLPhotographicImageStorage
SegmentationStorage
RTIonPlanStorage
XRay3DAngiographicImageStorage
EnhancedXAImageStorage
RTIonBeamsTreatmentRecordStorage
SurfaceSegmentationStorage
VLWholeSlideMicroscopyImageStorage
MS_END

Examples:

GenerateStandardSOPClasses.cxx.

25.164.2.2 enum gdcm::MediaStorage::ObjectType

Enumerator:

NoObject
Video
Waveform
Audio
PDF
URI
Segmentation
ObjectEnd

25.164.3 Constructor & Destructor Documentation

25.164.3.1 gdcm::MediaStorage::MediaStorage (MSType *type* = MS_END) [inline]

25.164.4 Member Function Documentation

25.164.4.1 const char* gdcm::MediaStorage::GetModality ( ) const

25.164.4.2 unsigned int gdcm::MediaStorage::GetModalityDimension ( ) const

25.164.4.3 static const char* gdcm::MediaStorage::GetMSString (MSType *ts*) [static]

Return the Media String associated. Will return NULL for MS_END.

Examples:

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

25.164.4.4 static MSType gdcm::MediaStorage::GetMSType (const char * *str*) [static]

Examples:

TestReader.cxx.

25.164.4.5 static unsigned int gdcm::MediaStorage::GetNumberOfModality ( ) [static]

25.164.4.6 static unsigned int gdcm::MediaStorage::GetNumberOfMSString ( ) [static]

25.164.4.7 static unsigned int gdcm::MediaStorage::GetNumberOfMSType ( ) [static]

25.164.4.8 const char* gdcm::MediaStorage::GetString ( ) const

Return the Media String of the object.

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GetSub-SequenceData.cxx, iU22tomultisc.cxx, and StreamImageReaderTest.cxx.

25.164.4.9 void gdcm::MediaStorage::GuessFromModality (const char * *modality,* unsigned int *dimension* = 2)

25.164.4.10 static bool gdcm::MediaStorage::IsImage (MSType *ts*) [static]

Returns whether DICOM has a Pixel Data element (7fe0,0010)

Warning

MRSpectroscopyStorage could be image but are not

25.164.4.11 bool gdcm::MediaStorage::IsUndefined ( ) const [inline]

Examples:

TestReader.cxx.

25.164.4.12 gdcm::MediaStorage::operator MSType ( ) const [inline]

25.164.4.13 bool gdcm::MediaStorage::SetFromDataSet (DataSet const & *ds*)

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

25.164.4.14 bool gdcm::MediaStorage::SetFromFile (File const & *file*)

Attempt to set the MediaStorage from a file: WARNING: When no MediaStorage & Modality are found BUT a PixelData element is found then MediaStorage is set to the default SecondaryCaptureImageStorage (return value is false in this case)

Examples:

gdcmrtionplan.cxx, gdcmrtplan.cxx, ReadAndDumpDICOMDIR.cxx, and TestReader.cxx.

25.164.4.15 bool gdcm::MediaStorage::SetFromHeader (FileMetaInformation const & *fmi*)

25.164.4.16 bool gdcm::MediaStorage::SetFromModality (DataSet const & *ds*)

25.164.4.17 void gdcm::MediaStorage::SetFromSourceImageSequence (DataSet const & *ds*) [protected]

25.164.5 Friends And Related Function Documentation

25.164.5.1 std::ostream& operator<< (std::ostream & *os*, const MediaStorage & *ms*) [friend]

The documentation for this class was generated from the following file:

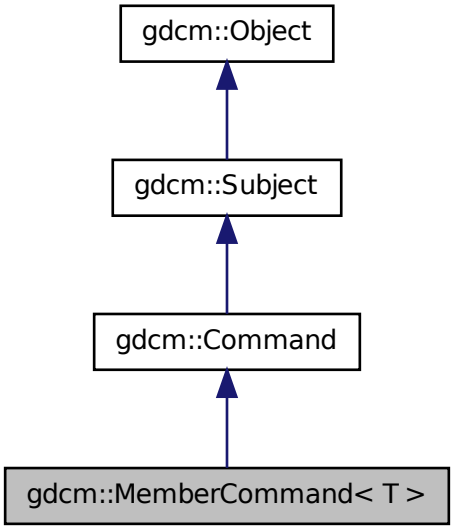
- gdcmMediaStorage.h

25.165 gdcm::MemberCommand< T > Class Template Reference

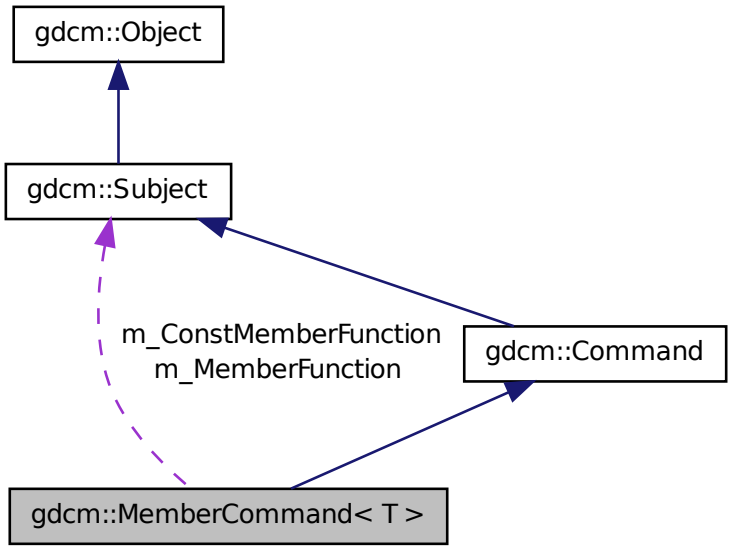
Command subclass that calls a pointer to a member function.

`#include <gdcmCommand.h>`

Inheritance diagram for gdcm::MemberCommand< T >:



Collaboration diagram for gdcm::MemberCommand< T >:



Public Types

- typedef MemberCommand Self
- typedef void(T::∗ TConstMemberFunctionPointer)(const Subject ∗, const Event &)
- typedef void(T::∗ TMemberFunctionPointer)(Subject ∗, const Event &)

Public Member Functions

- virtual void Execute (Subject ∗caller, const Event &event)
- virtual void Execute (const Subject ∗caller, const Event &event)
- void SetCallbackFunction (T ∗object, TMemberFunctionPointer memberFunction)
- void SetCallbackFunction (T ∗object, TConstMemberFunctionPointer memberFunction)

Static Public Member Functions

- static SmartPointer
  < MemberCommand > New ()

Protected Member Functions

- MemberCommand ()
- virtual ∼MemberCommand ()

Protected Attributes

- TConstMemberFunctionPointer m_ConstMemberFunction
- TMemberFunctionPointer m_MemberFunction
- T ∗ m_This

25.165.1 Detailed Description

template<class T>class gdcm::MemberCommand< T >

Command subclass that calls a pointer to a member function.

MemberCommand calls a pointer to a member function with the same arguments as Execute on Command.

25.165.2 Member Typedef Documentation

25.165.2.1 template<class T > typedef MemberCommand gdcm::MemberCommand< T >::Self

Standard class typedefs.

25.165.2.2 template<class T > typedef void(T::∗ gdcm::MemberCommand< T >::TConstMemberFunctionPointer)(const Subject ∗, const Event &)

25.165.2.3 template<class T > typedef void(T::∗ gdcm::MemberCommand< T >::TMemberFunctionPointer)(Subject ∗, const Event &)

pointer to a member function that takes a Subject∗ and the event

25.165.3 Constructor & Destructor Documentation

25.165.3.1 template<class T > gdcm::MemberCommand< T >::MemberCommand ( ) [inline], [protected]

Referenced by gdcm::MemberCommand< T >::New().

25.165.3.2 template<class T > virtual gdcm::MemberCommand< T >::∼MemberCommand ( ) [inline], [protected], [virtual]

25.165.4 Member Function Documentation

25.165.4.1 template<class T > virtual void gdcm::MemberCommand< T >::Execute (Subject ∗ *caller,* const Event & *event*) [inline], [virtual]

Invoke the member function.

Implements gdcm::Command.

References gdcm::MemberCommand< T >::m_MemberFunction.

25.165.4.2 template<class T> virtual void gdcm::MemberCommand< T >::Execute (const Subject * *caller,* const Event & *event*) [inline],[virtual]

Invoke the member function with a const object.

Implements gdcm::Command.

References gdcm::MemberCommand< T >::m_ConstMemberFunction.

25.165.4.3 template<class T> static SmartPointer<MemberCommand> gdcm::MemberCommand< T >::New ( ) [inline],[static]

Method for creation through the object factory.

References gdcm::MemberCommand< T >::MemberCommand().

25.165.4.4 template<class T> void gdcm::MemberCommand< T >::SetCallbackFunction (T * *object,* TMemberFunctionPointer *memberFunction*) [inline]

Run-time type information (and related methods). Set the callback function along with the object that it will be invoked on.

References gdcm::MemberCommand< T >::m_MemberFunction, and gdcm::MemberCommand< T >::m_This.

25.165.4.5 template<class T> void gdcm::MemberCommand< T >::SetCallbackFunction (T * *object,* TConstMemberFunctionPointer *memberFunction*) [inline]

References gdcm::MemberCommand< T >::m_ConstMemberFunction, and gdcm::MemberCommand< T >::m_This.

25.165.5 Member Data Documentation

25.165.5.1 template<class T> TConstMemberFunctionPointer gdcm::MemberCommand< T >::m_ConstMemberFunction [protected]

Referenced by gdcm::MemberCommand< T >::Execute(), and gdcm::MemberCommand< T >::SetCallbackFunction().

25.165.5.2 template<class T> TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction [protected]

Referenced by gdcm::MemberCommand< T >::Execute(), and gdcm::MemberCommand< T >::SetCallbackFunction().

25.165.5.3 template<class T> T* gdcm::MemberCommand< T >::m_This [protected]

Referenced by gdcm::MemberCommand< T >::SetCallbackFunction().

The documentation for this class was generated from the following file:

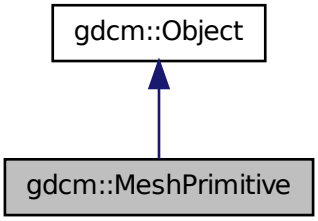
- gdcmCommand.h

25.166 gdcm::MeshPrimitive Class Reference

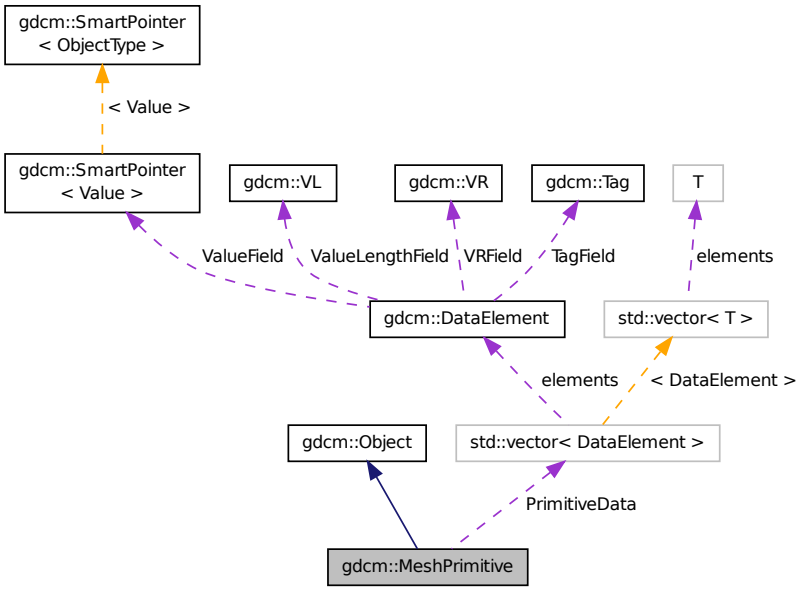
This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

`#include <gdcmMeshPrimitive.h>`

Inheritance diagram for gdcm::MeshPrimitive:



Collaboration diagram for gdcm::MeshPrimitive:



Public Types

- enum MPType {
  VERTEX = 0,
  EDGE,
  TRIANGLE,
  TRIANGLE_STRIP,
  TRIANGLE_FAN,
  LINE,
  FACET,
  MPType_END }

  This enumeration defines primitive types.
- typedef std::vector< DataElement > PrimitivesData

Public Member Functions

- MeshPrimitive ()
- virtual ~MeshPrimitive ()
- void AddPrimitiveData (DataElement const &de)
- unsigned int GetNumberOfPrimitivesData () const
- const DataElement & GetPrimitiveData () const
- DataElement & GetPrimitiveData ()
- const DataElement & GetPrimitiveData (const unsigned int idx) const
- DataElement & GetPrimitiveData (const unsigned int idx)
- const PrimitivesData & GetPrimitivesData () const
- PrimitivesData & GetPrimitivesData ()
- MPType GetPrimitiveType () const
- void SetPrimitiveData (DataElement const &de)
- void SetPrimitiveData (const unsigned int idx, DataElement const &de)
- void SetPrimitivesData (PrimitivesData const &DEs)
- void SetPrimitiveType (const MPType type)

Static Public Member Functions

- static MPType GetMPType (const char *type)
- static const char * GetMPTypeString (const MPType type)

Protected Attributes

- PrimitivesData PrimitiveData
- MPType PrimitiveType

Additional Inherited Members

25.166.1 Detailed Description

This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

25.166.2 Member Typedef Documentation

25.166.2.1 typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData

25.166.3 Member Enumeration Documentation

25.166.3.1 enum gdcm::MeshPrimitive::MPType

This enumeration defines primitive types.

See also

PS 3.3 C.27.4.1

Enumerator:

VERTEX

EDGE

TRIANGLE

TRIANGLE_STRIP

TRIANGLE_FAN

LINE

FACET

MPType_END

25.166.4 Constructor & Destructor Documentation

25.166.4.1 gdcm::MeshPrimitive::MeshPrimitive ( )

25.166.4.2 virtual gdcm::MeshPrimitive::~MeshPrimitive ( ) [virtual]

25.166.5 Member Function Documentation

25.166.5.1 void gdcm::MeshPrimitive::AddPrimitiveData (DataElement const & *de*)

25.166.5.2 static MPType gdcm::MeshPrimitive::GetMPType (const char * *type*) [static]

25.166.5.3 static const char* gdcm::MeshPrimitive::GetMPTypeString (const MPType *type*) [static]

25.166.5.4 unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData ( ) const

25.166.5.5 const DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( ) const

25.166.5.6 DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( )

25.166.5.7 const DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int *idx*) const

25.166.5.8 DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int *idx*)

25.166.5.9 const PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( ) const

25.166.5.10 PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( )

25.166.5.11 MPType gdcm::MeshPrimitive::GetPrimitiveType ( ) const

25.166.5.12 void gdcm::MeshPrimitive::SetPrimitiveData (DataElement const & *de*)

25.166.5.13 void gdcm::MeshPrimitive::SetPrimitiveData (const unsigned int *idx*, DataElement const & *de*)

25.166.5.14 void gdcm::MeshPrimitive::SetPrimitivesData (PrimitivesData const & *DEs*)

25.166.5.15 void gdcm::MeshPrimitive::SetPrimitiveType (const MPType *type*)

25.166.6 Member Data Documentation

25.166.6.1 PrimitivesData gdcm::MeshPrimitive::PrimitiveData [protected]

25.166.6.2 MPType gdcm::MeshPrimitive::PrimitiveType [protected]

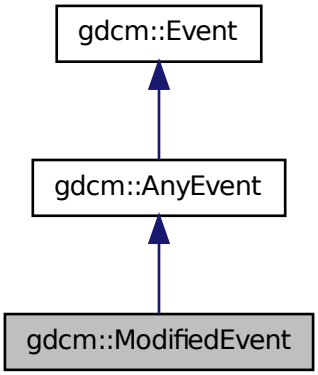
The documentation for this class was generated from the following file:

- gdcmMeshPrimitive.h

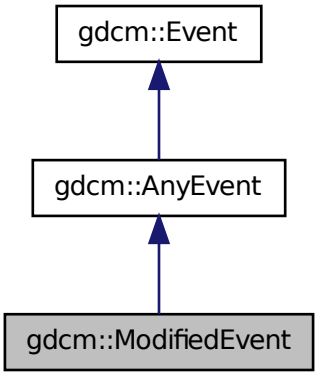
25.167 gdcm::ModifiedEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::ModifiedEvent:



Collaboration diagram for gdcm::ModifiedEvent:



The documentation for this class was generated from the following file:

- gdcmEvent.h

25.168 gdcm::Module Class Reference

Class for representing a Module.

`#include <gdcmModule.h>`

Public Types

- typedef std::vector< std::string > ArrayIncludeMacrosType
- typedef std::map< Tag,
  ModuleEntry > MapModuleEntry

Public Member Functions

- Module ()
- void AddMacro (const char *include)
- void AddModuleEntry (const Tag &tag, const ModuleEntry &module)

  Will add a ModuleEntry direcly at root-level. See Macro for nested-included level.
- void Clear ()
- bool FindModuleEntryInMacros (Macros const &macros, const Tag &tag) const
- const ModuleEntry & GetModuleEntryInMacros (Macros const &macros, const Tag &tag) const
- const char * GetName () const
- void SetName (const char *name)
- bool Verify (const DataSet &ds, Usage const &usage) const

Friends

- std::ostream & operator<< (std::ostream &_os, const Module &_val)

25.168.1 Detailed Description

Class for representing a Module.

Note

Module: A set of Attributes within an Information Entity or Normalized IOD which are logically related to each other.

See also

Macro

Examples:

TraverseModules.cxx.

25.168.2 Member Typedef Documentation

25.168.2.1 typedef std::vector<std::string> gdcm::Module::ArrayIncludeMacrosType

25.168.2.2 typedef std::map<Tag, ModuleEntry> gdcm::Module::MapModuleEntry

25.168.3 Constructor & Destructor Documentation

25.168.3.1 gdcm::Module::Module ( ) [inline]

25.168.4 Member Function Documentation

25.168.4.1 void gdcm::Module::AddMacro (const char * *include*) [inline]

25.168.4.2 void gdcm::Module::AddModuleEntry (const Tag & *tag,* const ModuleEntry & *module*) [inline]

Will add a ModuleEntry direcly at root-level. See Macro for nested-included level.

25.168.4.3 void gdcm::Module::Clear ( ) [inline]

25.168.4.4 bool gdcm::Module::FindModuleEntryInMacros (Macros const & *macros,* const Tag & *tag*) const

Find or Get a ModuleEntry. ModuleEntry are either search are root-level or within nested-macro included in module.

Examples:

TraverseModules.cxx.

25.168.4.5 const ModuleEntry& gdcm::Module::GetModuleEntryInMacros (Macros const & *macros,* const Tag & *tag*) const

Examples:

TraverseModules.cxx.

25.168.4.6 const char* gdcm::Module::GetName ( ) const [inline]

25.168.4.7 void gdcm::Module::SetName (const char * *name*) [inline]

25.168.4.8 bool gdcm::Module::Verify (const **DataSet** & *ds*, **Usage** const & *usage*) const

25.168.5 Friends And Related Function Documentation

25.168.5.1 std::ostream& operator<< (std::ostream & *_os*, const **Module** & *_val*) [friend]

The documentation for this class was generated from the following file:

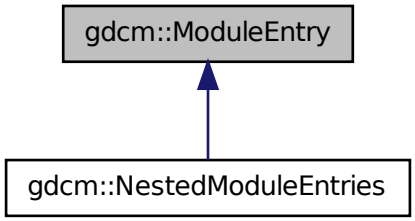
- gdcmModule.h

25.169 gdcm::ModuleEntry Class Reference

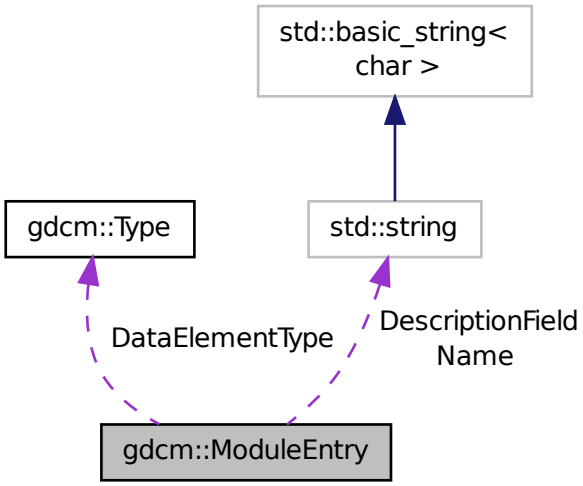
Class for representing a ModuleEntry.

`#include <gdcmModuleEntry.h>`

Inheritance diagram for gdcm::ModuleEntry:



Collaboration diagram for gdcm::ModuleEntry:



Public Types

- typedef std::string Description

Public Member Functions

- ModuleEntry (const char ∗name="", const char ∗type="3", const char ∗description="")
- virtual ∼ModuleEntry ()
- const Description & GetDescription () const
- const char ∗ GetName () const
- const Type & GetType () const
- void SetDescription (const char ∗d)
- void SetName (const char ∗name)
- void SetType (const Type &type)

Protected Attributes

- Type DataElementType
- Description DescriptionField
- std::string Name

Friends

- std::ostream & operator<< (std::ostream &_os, const ModuleEntry &_val)

25.169.1 Detailed Description

Class for representing a ModuleEntry.

Note

bla

See also

DictEntry

Examples:

TraverseModules.cxx.

25.169.2 Member Typedef Documentation

25.169.2.1 typedef std::string gdcm::ModuleEntry::Description

25.169.3 Constructor & Destructor Documentation

25.169.3.1 gdcm::ModuleEntry::ModuleEntry (const char * *name* = " ", const char * *type* = "3", const char * *description* = " ") [inline]

References gdcm::Type::GetTypeType().

25.169.3.2 virtual gdcm::ModuleEntry::~ModuleEntry ( ) [inline], [virtual]

25.169.4 Member Function Documentation

25.169.4.1 const Description& gdcm::ModuleEntry::GetDescription ( ) const [inline]

25.169.4.2 const char* gdcm::ModuleEntry::GetName ( ) const [inline]

25.169.4.3 const Type& gdcm::ModuleEntry::GetType ( ) const [inline]

Examples:

TraverseModules.cxx.

25.169.4.4 void gdcm::ModuleEntry::SetDescription (const char * *d*) [inline]

25.169.4.5 void gdcm::ModuleEntry::SetName (const char * *name*) [inline]

25.169.4.6 void gdcm::ModuleEntry::SetType (const Type & *type*) [inline]

25.169.5 Friends And Related Function Documentation

25.169.5.1 std::ostream& operator<< (std::ostream & *_os*, const ModuleEntry & *_val*) [friend]

25.169.6 Member Data Documentation

25.169.6.1 Type gdcm::ModuleEntry::DataElementType [protected]

Referenced by gdcm::operator<<().

25.169.6.2 Description gdcm::ModuleEntry::DescriptionField [protected]

Referenced by gdcm::operator<<().

25.169.6.3 std::string gdcm::ModuleEntry::Name [protected]

Referenced by gdcm::operator<<().

The documentation for this class was generated from the following file:

- gdcmModuleEntry.h

25.170 gdcm::Modules Class Reference

Class for representing a Modules.

`#include <gdcmModules.h>`

Public Types

- typedef std::map< std::string, Module > ModuleMapType

Public Member Functions

- Modules ()
- void AddModule (const char *ref, const Module &module)
- void Clear ()
- const Module & GetModule (const char *name) const
- bool IsEmpty () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Modules &_val)

25.170.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples:

TraverseModules.cxx.

25.170.2 Member Typedef Documentation

25.170.2.1 typedef std::map<std::string, Module> gdcm::Modules::ModuleMapType

25.170.3 Constructor & Destructor Documentation

25.170.3.1 gdcm::Modules::Modules ( ) [inline]

25.170.4 Member Function Documentation

25.170.4.1 void gdcm::Modules::AddModule (const char * *ref*, const Module & *module*) [inline]

25.170.4.2 void gdcm::Modules::Clear ( ) [inline]

25.170.4.3 const Module& gdcm::Modules::GetModule (const char * *name*) const [inline]

25.170.4.4 bool gdcm::Modules::IsEmpty ( ) const [inline]

25.170.5 Friends And Related Function Documentation

25.170.5.1 std::ostream& operator<< (std::ostream & *_os*, const Modules & *_val*) [friend]

The documentation for this class was generated from the following file:

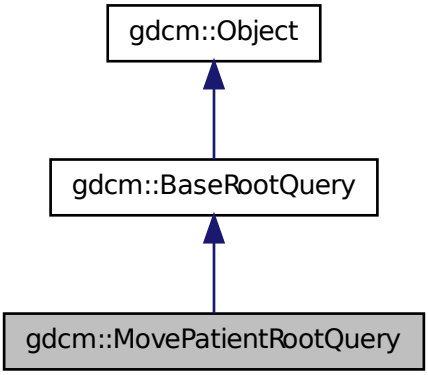
- gdcmModules.h

25.171 gdcm::MovePatientRootQuery Class Reference

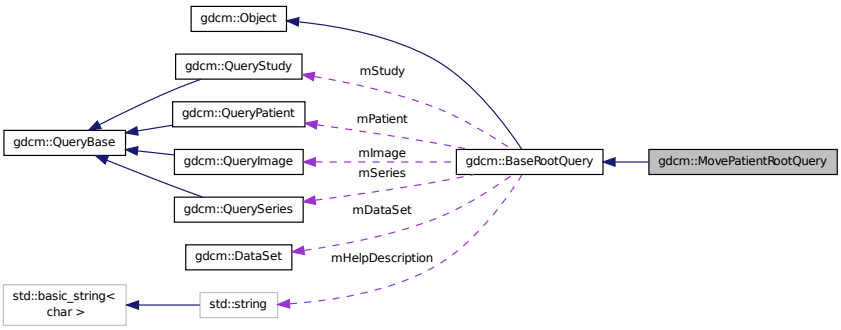
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.

```
#include <gdcmMovePatientRootQuery.h>
```

Inheritance diagram for gdcm::MovePatientRootQuery:



Collaboration diagram for gdcm::MovePatientRootQuery:



Public Member Functions

- MovePatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

25.171.1 Detailed Description

MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.

25.171.2 Constructor & Destructor Documentation

25.171.2.1 gdcm::MovePatientRootQuery::MovePatientRootQuery ( )

25.171.3 Member Function Documentation

25.171.3.1 UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseRootQuery.

25.171.3.2 std::vector<Tag> gdcm::MovePatientRootQuery::GetTagListByLevel (const EQueryLevel & *inQueryLevel*) [virtual]

this function will return all tags at a given query level, so that ∗they maybe selected for searching. The boolean forFind is true ∗if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

25.171.3.3 void gdcm::MovePatientRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [virtual]

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

25.171.3.4 bool gdcm::MovePatientRootQuery::ValidateQuery (bool *inStrict* = true) const [virtual]

have to be able to ensure that ∗0x8,0x52 is set (which will be true if InitializeDataSet is called...) ∗that the level is appropriate (ie, not setting PATIENT for a study query ∗that the tags in the query match the right level (either required, unique, optional) ∗by default, this function checks to see if the query is for finding, which is more ∗permissive than for moving. For moving, only the unique tags are allowed. ∗10 Jan 2011: adding in the 'strict' mode. ∗according to the standard (at least, how I've read it), only tags for a particular ∗level should be allowed in a particular query (ie, just series level tags in a series ∗level query). However, it seems that dcm4chee doesn't share that interpretation. ∗So, if 'inStrict' is false, then tags from the current level and all higher levels ∗are now considered valid. So, if you're doing a non-strict series-level query, ∗tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

25.171.4 Friends And Related Function Documentation

25.171.4.1 friend class QueryFactory [friend]

Reimplemented from gdcm::BaseRootQuery.

The documentation for this class was generated from the following file:

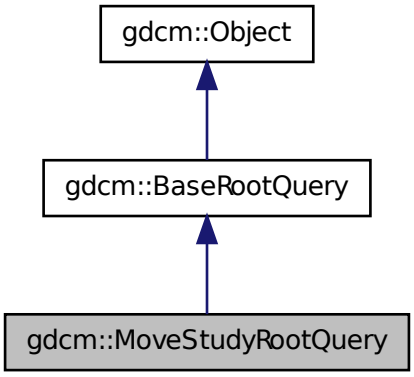
- gdcmMovePatientRootQuery.h

25.172 gdcm::MoveStudyRootQuery Class Reference

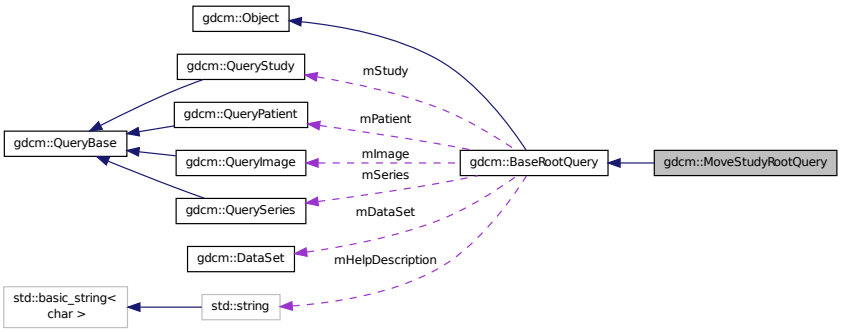
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

```
#include <gdcmMoveStudyRootQuery.h>
```

Inheritance diagram for gdcm::MoveStudyRootQuery:



Collaboration diagram for gdcm::MoveStudyRootQuery:



Public Member Functions

- MoveStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)

- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

25.172.1 Detailed Description

MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

25.172.2 Constructor & Destructor Documentation

25.172.2.1 gdcm::MoveStudyRootQuery::MoveStudyRootQuery ( )

25.172.3 Member Function Documentation

25.172.3.1 UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseRootQuery.

25.172.3.2 std::vector<Tag> gdcm::MoveStudyRootQuery::GetTagListByLevel (const EQueryLevel & *inQueryLevel*) [virtual]

this function will return all tags at a given query level, so that ∗they maybe selected for searching. The boolean forFind is true ∗if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

25.172.3.3 void gdcm::MoveStudyRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [virtual]

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

25.172.3.4 bool gdcm::MoveStudyRootQuery::ValidateQuery (bool *inStrict* = true) const [virtual]

have to be able to ensure that ∗0x8,0x52 is set (which will be true if InitializeDataSet is called...) ∗that the level is appropriate (ie, not setting PATIENT for a study query ∗that the tags in the query match the right level (either required, unique, optional) ∗by default, this function checks to see if the query is for finding, which is more ∗permissive than for moving. For moving, only the unique tags are allowed. ∗10 Jan 2011: adding in the 'strict' mode. ∗according to the standard (at least, how I've read it), only tags for a particular ∗level should be allowed in a particular query (ie, just series level tags in a series ∗level query). However, it seems that dcm4chee doesn't share that interpretation. ∗So, if 'inStrict' is false, then tags from the current level and all higher levels ∗are now considered valid. So, if you're doing a non-strict series-level query, ∗tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

25.172.4 Friends And Related Function Documentation

25.172.4.1 friend class QueryFactory [friend]

Reimplemented from gdcm::BaseRootQuery.

The documentation for this class was generated from the following file:

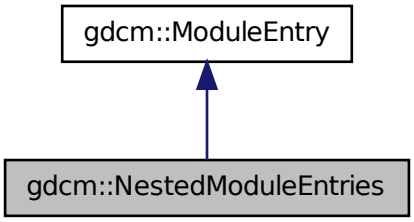
- gdcmMoveStudyRootQuery.h

25.173 gdcm::NestedModuleEntries Class Reference

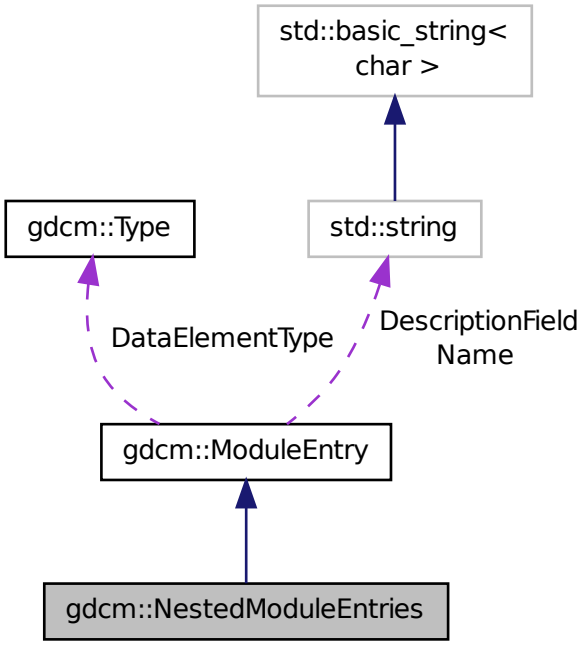
Class for representing a NestedModuleEntries.

`#include <gdcmNestedModuleEntries.h>`

Inheritance diagram for gdcm::NestedModuleEntries:



Collaboration diagram for gdcm::NestedModuleEntries:



Public Types

- typedef std::vector
  < ModuleEntry >::size_type SizeType

Public Member Functions

- NestedModuleEntries (const char *name="", const char *type="3", const char *description="")
- void AddModuleEntry (const ModuleEntry &me)
- const ModuleEntry & GetModuleEntry (SizeType idx) const
- ModuleEntry & GetModuleEntry (SizeType idx)
- SizeType GetNumberOfModuleEntries ()

Friends

- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)

Additional Inherited Members

25.173.1 Detailed Description

Class for representing a NestedModuleEntries.

Note

bla

See also

ModuleEntry

25.173.2 Member Typedef Documentation

25.173.2.1 typedef std::vector<ModuleEntry>::size_type gdcm::NestedModuleEntries::SizeType

25.173.3 Constructor & Destructor Documentation

25.173.3.1 gdcm::NestedModuleEntries::NestedModuleEntries (const char * *name* = " ", const char * *type* = "3", const char * *description* = " ") [inline]

25.173.4 Member Function Documentation

25.173.4.1 void gdcm::NestedModuleEntries::AddModuleEntry (const ModuleEntry & *me*) [inline]

25.173.4.2 const ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (SizeType *idx*) const [inline]

25.173.4.3 ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (SizeType *idx*) [inline]

25.173.4.4 SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries ( ) [inline]

25.173.5 Friends And Related Function Documentation

25.173.5.1 std::ostream& operator<< (std::ostream & *_os*, const NestedModuleEntries & *_val*) [friend]

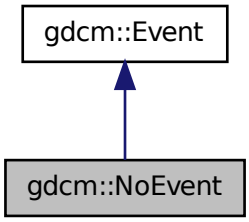
The documentation for this class was generated from the following file:

- gdcmNestedModuleEntries.h

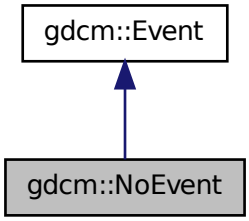
25.174 gdcm::NoEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



Additional Inherited Members

25.174.1 Detailed Description

Define some common GDCM events

The documentation for this class was generated from the following file:

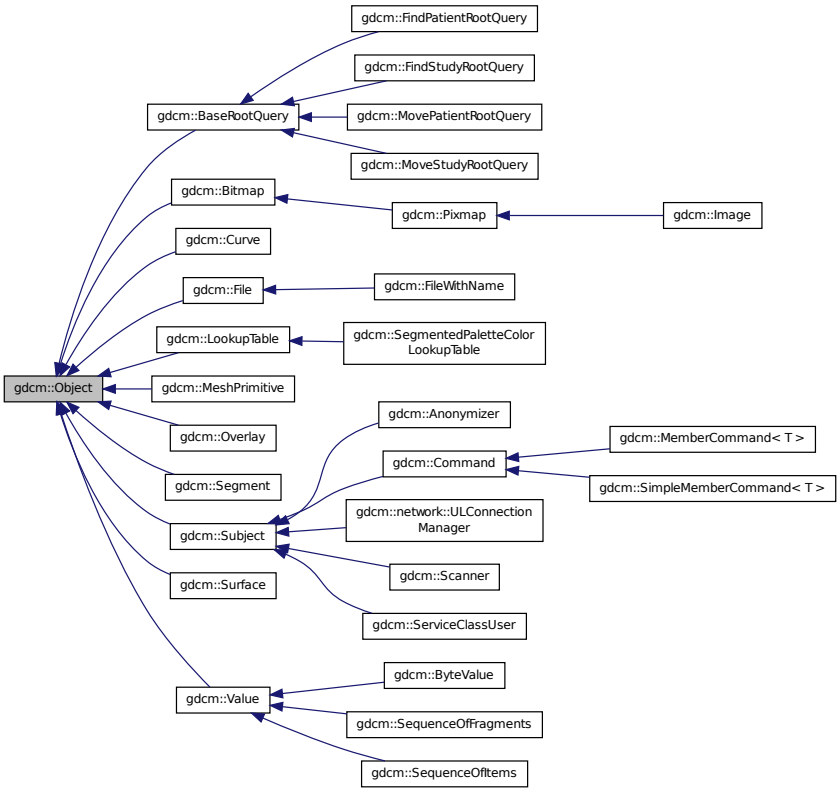
- gdcmEvent.h

25.175 gdcm::Object Class Reference

Object.

```
#include <gdcmObject.h>
```

Inheritance diagram for gdcm::Object:



Public Member Functions

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assigment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Protected Member Functions

- void Register ()
- void UnRegister ()

Friends

- std::ostream & operator<< (std::ostream &os, const Object &obj)
- class SmartPointer

25.175.1 Detailed Description

Object.

Note

main superclass for object that want to use SmartPointer invasive ref counting system

See also

SmartPointer

25.175.2 Constructor & Destructor Documentation

25.175.2.1 gdcm::Object::Object ( ) [inline]

25.175.2.2 virtual gdcm::Object::~Object ( ) [inline], [virtual]

25.175.2.3 gdcm::Object::Object (const Object &) [inline]

Special requirement for copy/cstor, assigment operator.

25.175.3 Member Function Documentation

25.175.3.1 void gdcm::Object::operator= (const Object &) [inline]

25.175.3.2 virtual void gdcm::Object::Print (std::ostream &) const [inline], [virtual]

Reimplemented in gdcm::ByteValue, gdcm::SequenceOfItems, gdcm::SequenceOfFragments, gdcm::Scanner, gdcm::-Image, gdcm::Overlay, gdcm::Bitmap, gdcm::Curve, gdcm::LookupTable, gdcm::Pixmap, and gdcm::SegmentedPalette-ColorLookupTable.

Examples:

ReadAndDumpDICOMDIR.cxx.

Referenced by gdcm::operator<<().

25.175.3.3 void gdcm::Object::Register ( ) [inline], [protected]

25.175.3.4 void gdcm::Object::UnRegister ( ) [inline], [protected]

25.175.4 Friends And Related Function Documentation

25.175.4.1 std::ostream& operator<< (std::ostream & *os*, const Object & *obj*) [friend]

25.175.4.2 friend class SmartPointer [friend]

The documentation for this class was generated from the following file:

- gdcmObject.h

25.176 gdcm::OneShotReadBuf Struct Reference

```
#include <gdcmStreamImageReader.h>
```

Public Member Functions

- OneShotReadBuf (void ∗s, std::size_t n)

25.176.1 Constructor & Destructor Documentation

25.176.1.1 gdcm::OneShotReadBuf::OneShotReadBuf (void ∗ *s*, std::size_t *n*) [inline]

The documentation for this struct was generated from the following file:

- gdcmStreamImageReader.h

25.177 gdcm::Orientation Class Reference

class to handle Orientation

```
#include <gdcmOrientation.h>
```

Public Types

- enum OrientationType {
  UNKNOWN,
  AXIAL,
  CORONAL,
  SAGITTAL,
  OBLIQUE }

Public Member Functions

- Orientation ()
- ∼Orientation ()
- void Print (std::ostream &) const

  Print.

Static Public Member Functions

- static const char ∗ GetLabel (OrientationType type)

  Return the label of an Orientation.
- static double GetObliquityThresholdCosineValue ()
- static OrientationType GetType (const double dircos[6])
- static void SetObliquityThresholdCosineValue (double val)

  ObliquityThresholdCosineValue stuff.

Static Protected Member Functions

- static char GetMajorAxisFromPatientRelativeDirectionCosine (double x, double y, double z)

Friends

- std::ostream & operator<< (std::ostream &_os, const Orientation &o)

25.177.1 Detailed Description

class to handle Orientation

25.177.2 Member Enumeration Documentation

25.177.2.1 enum gdcm::Orientation::OrientationType

Enumerator:

UNKNOWN

AXIAL

CORONAL

SAGITTAL

OBLIQUE

25.177.3 Constructor & Destructor Documentation

25.177.3.1 gdcm::Orientation::Orientation ( )

25.177.3.2 gdcm::Orientation::~Orientation ( )

25.177.4 Member Function Documentation

25.177.4.1 static const char* gdcm::Orientation::GetLabel (OrientationType *type*) [static]

Return the label of an Orientation.

25.177.4.2 static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (double *x*, double *y*, double *z*) [static], [protected]

25.177.4.3 static double gdcm::Orientation::GetObliquityThresholdCosineValue ( ) [static]

25.177.4.4 static OrientationType gdcm::Orientation::GetType (const double *dircos[6]*) [static]

Return the type of orientation from a direction cosines Input is an array of 6 double

25.177.4.5 void gdcm::Orientation::Print (std::ostream &) const

Print.

Referenced by gdcm::operator<<().

25.177.4.6 static void gdcm::Orientation::SetObliquityThresholdCosineValue (double *val*) [static]

ObliquityThresholdCosineValue stuff.

25.177.5 Friends And Related Function Documentation

25.177.5.1 std::ostream& operator<< (std::ostream & *_os*, const Orientation & *o*) [friend]

The documentation for this class was generated from the following file:

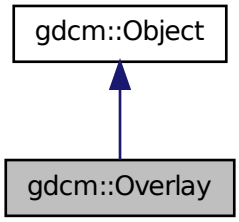
- gdcmOrientation.h

25.178 gdcm::Overlay Class Reference

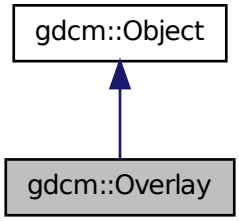
Overlay class.

`#include <gdcmOverlay.h>`

Inheritance diagram for gdcm::Overlay:



Collaboration diagram for gdcm::Overlay:



Public Member Functions

- Overlay ()
- Overlay (Overlay const &ov)
- ∼Overlay ()
- void Decode (std::istream &is, std::ostream &os)
- void Decompress (std::ostream &os) const
- unsigned short GetBitPosition () const

  return bit position
- unsigned short GetBitsAllocated () const

  return bits allocated
- bool GetBuffer (char ∗buffer) const
- unsigned short GetColumns () const

  get columns
- const char ∗ GetDescription () const

  get description
- unsigned short GetGroup () const

  Get Group number.
- const signed short ∗ GetOrigin () const

  get origin
- const ByteValue & GetOverlayData () const
- unsigned short GetRows () const

  get rows
- const char ∗ GetType () const

  get type
- bool GetUnpackBuffer (unsigned char ∗buffer) const
- bool GrabOverlayFromPixelData (DataSet const &ds)
- bool IsEmpty () const
- bool IsInPixelData () const
- void IsInPixelData (bool b)
- bool IsZero () const

  return true if all bits are set to 0

- void Print (std::ostream &) const

  Print.
- void SetBitPosition (unsigned short bitposition)

  set bit position
- void SetBitsAllocated (unsigned short bitsallocated)

  set bits allocated
- void SetColumns (unsigned short columns)

  set columns
- void SetDescription (const char *description)

  set description
- void SetFrameOrigin (unsigned short frameorigin)

  set frame origin
- void SetGroup (unsigned short group)

  Set Group number.
- void SetNumberOfFrames (unsigned int numberofframes)

  set number of frames
- void SetOrigin (const signed short *origin)

  set origin
- void SetOverlay (const char *array, unsigned int length)

  set overlay from byte array + length
- void SetRows (unsigned short rows)

  set rows
- void SetType (const char *type)

  set type
- void Update (const DataElement &de)

  Update overlay from data element de:

Additional Inherited Members

25.178.1 Detailed Description

Overlay class.

Note

see AreOverlaysInPixelData

Todo Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differenciate them (other than the group tag).

Example:

25.178.2 Constructor & Destructor Documentation

25.178.2.1 gdcm::Overlay::Overlay ( )

25.178.2.2 gdcm::Overlay::~Overlay ( )

25.178.2.3 gdcm::Overlay::Overlay (Overlay const & *ov*)

25.178.3 Member Function Documentation

25.178.3.1 void gdcm::Overlay::Decode (std::istream & *is*, std::ostream & *os*)

25.178.3.2 void gdcm::Overlay::Decompress (std::ostream & *os*) const

25.178.3.3 unsigned short gdcm::Overlay::GetBitPosition ( ) const

return bit position

25.178.3.4 unsigned short gdcm::Overlay::GetBitsAllocated ( ) const

return bits allocated

25.178.3.5 bool gdcm::Overlay::GetBuffer (char * *buffer*) const

25.178.3.6 unsigned short gdcm::Overlay::GetColumns ( ) const

get columns

25.178.3.7 const char* gdcm::Overlay::GetDescription ( ) const

get description

25.178.3.8 unsigned short gdcm::Overlay::GetGroup ( ) const

Get Group number.

25.178.3.9 const signed short* gdcm::Overlay::GetOrigin ( ) const

get origin

25.178.3.10 const ByteValue& gdcm::Overlay::GetOverlayData ( ) const

25.178.3.11 unsigned short gdcm::Overlay::GetRows ( ) const

get rows

25.178.3.12 const char* gdcm::Overlay::GetType ( ) const

get type

25.178.3.13 bool gdcm::Overlay::GetUnpackBuffer (unsigned char * *buffer*) const

25.178.3.14 bool gdcm::Overlay::GrabOverlayFromPixelData (DataSet const & *ds*)

25.178.3.15 bool gdcm::Overlay::IsEmpty ( ) const

25.178.3.16 bool gdcm::Overlay::IsInPixelData ( ) const

25.178.3.17 void gdcm::Overlay::IsInPixelData (bool *b*)

25.178.3.18 bool gdcm::Overlay::IsZero ( ) const

return true if all bits are set to 0

25.178.3.19 void gdcm::Overlay::Print (std::ostream &) const [virtual]

Print.

Reimplemented from gdcm::Object.

25.178.3.20 void gdcm::Overlay::SetBitPosition (unsigned short *bitposition*)

set bit position

25.178.3.21 void gdcm::Overlay::SetBitsAllocated (unsigned short *bitsallocated*)

set bits allocated

25.178.3.22 void gdcm::Overlay::SetColumns (unsigned short *columns*)

set columns

25.178.3.23 void gdcm::Overlay::SetDescription (const char * *description*)

set description

25.178.3.24 void gdcm::Overlay::SetFrameOrigin (unsigned short *frameorigin*)

set frame origin

25.178.3.25 void gdcm::Overlay::SetGroup (unsigned short *group*)

Set Group number.

25.178.3.26 void gdcm::Overlay::SetNumberOfFrames (unsigned int *numberofframes*)

set number of frames

25.178.3.27 void gdcm::Overlay::SetOrigin (const signed short * *origin*)

set origin

25.178.3.28 void gdcm::Overlay::SetOverlay (const char * *array,* unsigned int *length*)

set overlay from byte array + length

25.178.3.29 void gdcm::Overlay::SetRows (unsigned short *rows*)

set rows

25.178.3.30 void gdcm::Overlay::SetType (const char * *type*)

set type

25.178.3.31 void gdcm::Overlay::Update (const DataElement & *de*)

Update overlay from data element de:

The documentation for this class was generated from the following file:

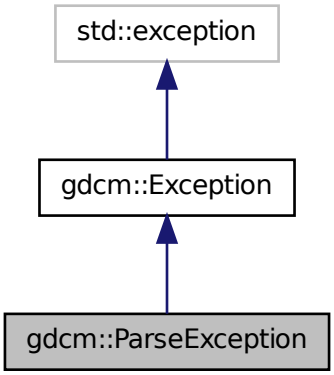
- gdcmOverlay.h

25.179 gdcm::ParseException Class Reference

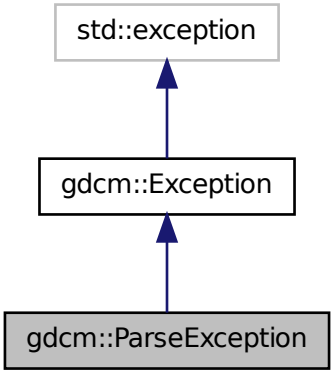
ParseException Standard exception handling object.

```
#include <gdcmParseException.h>
```

Inheritance diagram for gdcm::ParseException:



Collaboration diagram for gdcm::ParseException:



Public Member Functions

- ParseException ()
- virtual ~ParseException () throw ()
- const DataElement & GetLastElement () const
- ParseException & operator= (const ParseException &orig)
- void SetLastElement (DataElement &de)

25.179.1 Detailed Description

ParseException Standard exception handling object.

25.179.2 Constructor & Destructor Documentation

25.179.2.1 gdcm::ParseException::ParseException ( ) [inline]

25.179.2.2 virtual gdcm::ParseException::∼ParseException ( ) throw () [inline], [virtual]

25.179.3 Member Function Documentation

25.179.3.1 const DataElement& gdcm::ParseException::GetLastElement ( ) const [inline]

25.179.3.2 ParseException& gdcm::ParseException::operator= (const ParseException & *orig*) [inline]

Assignment operator.

25.179.3.3 void gdcm::ParseException::SetLastElement (DataElement & *de*) [inline]

Equivalence operator.

Referenced by gdcm::Fragment::ReadValue().

The documentation for this class was generated from the following file:

- gdcmParseException.h

25.180 gdcm::Parser Class Reference

Parser ala XML_Parser from expat (SAX)

```
#include <gdcmParser.h>
```

Public Types

- typedef void(∗ EndElementHandler)(void ∗userData, const Tag &name)
- enum ErrorType {
  NoError,
  NoMemoryError,
  SyntaxError,
  NoElementsError,
  TagMismatchError,
  DuplicateAttributeError,
  JunkAfterDocElementError,
  UndefinedEntityError,
  UnexpectedStateError }
- typedef void(∗ StartElementHandler)(void ∗userData, const Tag &tag, const char ∗atts[])

Public Member Functions

- Parser ()
- ∼Parser ()
- unsigned long GetCurrentByteIndex () const
- ErrorType GetErrorCode () const
- void ∗ GetUserData () const
- bool Parse (const char ∗s, int len, bool isFinal)
- void SetElementHandler (StartElementHandler start, EndElementHandler end)
- void SetUserData (void ∗userData)

Static Public Member Functions

- static const char ∗ GetErrorString (ErrorType const &err)

Protected Member Functions

- char ∗ GetBuffer (int len)
- bool ParseBuffer (int len, bool isFinal)
- ErrorType Process ()

25.180.1 Detailed Description

Parser ala XML_Parser from expat (SAX)

Detailled description here

Note

Simple API for DICOM

25.180.2 Member Typedef Documentation

25.180.2.1 typedef void(∗ gdcm::Parser::EndElementHandler)(void ∗userData, const Tag &name)

25.180.2.2 typedef void(∗ gdcm::Parser::StartElementHandler)(void ∗userData, const Tag &tag, const char ∗atts[ ])

25.180.3 Member Enumeration Documentation

25.180.3.1 enum gdcm::Parser::ErrorType

Enumerator:

NoError

NoMemoryError

SyntaxError

NoElementsError

TagMismatchError

DuplicateAttributeError

JunkAfterDocElementError

UndefinedEntityError

UnexpectedStateError

25.180.4 Constructor & Destructor Documentation

25.180.4.1 gdcm::Parser::Parser ( ) [inline]

25.180.4.2 gdcm::Parser::∼Parser ( ) [inline]

25.180.5 Member Function Documentation

25.180.5.1 char∗ gdcm::Parser::GetBuffer (int *len*) [protected]

25.180.5.2 unsigned long gdcm::Parser::GetCurrentByteIndex ( ) const

25.180.5.3 ErrorType gdcm::Parser::GetErrorCode ( ) const

25.180.5.4 static const char∗ gdcm::Parser::GetErrorString (ErrorType const & *err*) [static]

25.180.5.5 void∗ gdcm::Parser::GetUserData ( ) const

25.180.5.6 bool gdcm::Parser::Parse (const char ∗ *s,* int *len,* bool *isFinal*)

25.180.5.7 bool gdcm::Parser::ParseBuffer (int *len,* bool *isFinal*) [protected]

25.180.5.8 ErrorType gdcm::Parser::Process ( ) [protected]

25.180.5.9 void gdcm::Parser::SetElementHandler (StartElementHandler *start,* EndElementHandler *end*)

25.180.5.10 void gdcm::Parser::SetUserData (void ∗ *userData*)

The documentation for this class was generated from the following file:

- gdcmParser.h

25.181 gdcm::Patient Class Reference

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

```
#include <gdcmPatient.h>
```

Public Member Functions

- Patient ()

25.181.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

25.181.2 Constructor & Destructor Documentation

25.181.2.1 gdcm::Patient::Patient ( ) [inline]

The documentation for this class was generated from the following file:

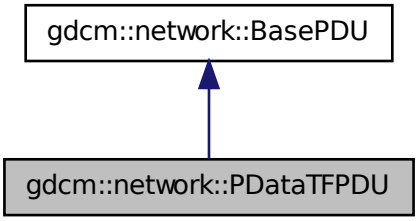
- gdcmPatient.h

25.182 gdcm::network::PDataTFPDU Class Reference

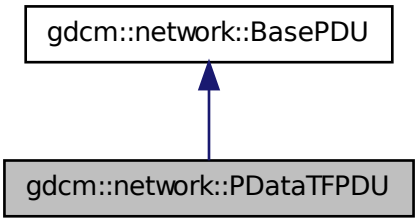
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

`#include <gdcmPDataTFPDU.h>`

Inheritance diagram for gdcm::network::PDataTFPDU:



Collaboration diagram for gdcm::network::PDataTFPDU:



Public Types

- typedef std::vector
  < PresentationDataValue >

::size_type SizeType

Public Member Functions

- PDataTFPDU ()
- void AddPresentationDataValue (PresentationDataValue const &pdv)
- SizeType GetNumberOfPresentationDataValues () const
- PresentationDataValue const & GetPresentationDataValue (SizeType i) const
- bool IsLastFragment () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

Protected Member Functions

- std::istream & ReadInto (std::istream &is, std::ostream &os)

25.182.1 Detailed Description

PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

25.182.2 Member Typedef Documentation

25.182.2.1 typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType

25.182.3 Constructor & Destructor Documentation

25.182.3.1 gdcm::network::PDataTFPDU::PDataTFPDU ( )

25.182.4 Member Function Documentation

25.182.4.1 void gdcm::network::PDataTFPDU::AddPresentationDataValue (PresentationDataValue const & *pdv*) [inline]

25.182.4.2 SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues ( ) const [inline]

25.182.4.3 PresentationDataValue const& gdcm::network::PDataTFPDU::GetPresentationDataValue (SizeType *i*) const [inline]

25.182.4.4 bool gdcm::network::PDataTFPDU::IsLastFragment ( ) const [virtual]

Implements gdcm::network::BasePDU.

25.182.4.5 void gdcm::network::PDataTFPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

25.182.4.6 std::istream& gdcm::network::PDataTFPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

25.182.4.7 std::istream& gdcm::network::PDataTFPDU::ReadInto (std::istream & *is*, std::ostream & *os*) [protected]

25.182.4.8 size_t gdcm::network::PDataTFPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

25.182.4.9 const std::ostream& gdcm::network::PDataTFPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

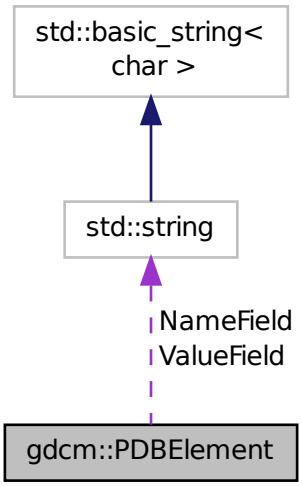
- gdcmPDataTFPDU.h

25.183 gdcm::PDBElement Class Reference

Class to represent a PDB Element.

`#include <gdcmPDBElement.h>`

Collaboration diagram for gdcm::PDBElement:



Public Member Functions

- PDBElement ()
- const char ∗ GetName () const

  Set/Get Name.
- const char ∗ GetValue () const

  Set/Get Value.
- bool operator== (const PDBElement &de) const
- void SetName (const char ∗name)
- void SetValue (const char ∗value)

Protected Attributes

- std::string NameField
- std::string ValueField

Friends

- std::ostream & operator<< (std::ostream &os, const PDBElement &val)

25.183.1 Detailed Description

Class to represent a PDB Element.

See also

PDBHeader

25.183.2 Constructor & Destructor Documentation

25.183.2.1 gdcm::PDBElement::PDBElement ( ) [inline]

25.183.3 Member Function Documentation

25.183.3.1 const char∗ gdcm::PDBElement::GetName ( ) const [inline]

Set/Get Name.

25.183.3.2 const char∗ gdcm::PDBElement::GetValue ( ) const [inline]

Set/Get Value.

25.183.3.3 bool gdcm::PDBElement::operator== (const PDBElement & *de*) const [inline]

References NameField, and ValueField.

25.183.3.4 void gdcm::PDBElement::SetName (const char * *name*) [inline]

25.183.3.5 void gdcm::PDBElement::SetValue (const char * *value*) [inline]

25.183.4 Friends And Related Function Documentation

25.183.4.1 std::ostream& operator<< (std::ostream & *os*, const **PDBElement** & *val*) [friend]

25.183.5 Member Data Documentation

25.183.5.1 std::string gdcm::PDBElement::NameField [protected]

Referenced by gdcm::operator<<(), and operator==().

25.183.5.2 std::string gdcm::PDBElement::ValueField [protected]

Referenced by gdcm::operator<<(), and operator==().

The documentation for this class was generated from the following file:

- gdcmPDBElement.h

25.184 gdcm::PDBHeader Class Reference

Class for PDBHeader.

```
#include <gdcmPDBHeader.h>
```

Public Member Functions

- PDBHeader ()
- ~PDBHeader ()
- bool FindPDBElementByName (const char *name)

  Return true if the PDB element matching name is found or not.
- const PDBElement & GetPDBElementByName (const char *name)
- bool LoadFromDataElement (DataElement const &de)

  Load the PDB Header from a DataElement of a DataSet.
- void Print (std::ostream &os) const

  Print.

Static Public Member Functions

- static const PrivateTag & GetPDBInfoTag ()

  Return the Private Tag where the PDB header is stored within a DICOM DataSet.

Protected Member Functions

- const PDBElement & GetPDBEEnd () const

Friends

- std::ostream & operator<< (std::ostream &_os, const PDBHeader &d)

25.184.1 Detailed Description

Class for PDBHeader.

GEMS MR Image have an Attribute (0025,1b,GEMS_SERS_01) which store the Acquisition parameter of the MR Image. It is compressed and can therefore not be used as is. This class de-encapsulated the Protocol Data Block and allow users to query element by name.

Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.
: the API of this class might change.

See also

CSAHeader

25.184.2 Constructor & Destructor Documentation

25.184.2.1 gdcm::PDBHeader::PDBHeader ( ) [inline]

25.184.2.2 gdcm::PDBHeader::~PDBHeader ( ) [inline]

25.184.3 Member Function Documentation

25.184.3.1 bool gdcm::PDBHeader::FindPDBElementByName (const char * *name*)

Return true if the PDB element matching name is found or not.

25.184.3.2 const PDBElement& gdcm::PDBHeader::GetPDBEEnd ( ) const [protected]

25.184.3.3 const PDBElement& gdcm::PDBHeader::GetPDBElementByName (const char * *name*)

Lookup in the PDB header if a PDB element match the name 'name':

Warning

Case Sensitive

25.184.3.4 static const PrivateTag& gdcm::PDBHeader::GetPDBInfoTag ( ) [static]

Return the Private Tag where the PDB header is stored within a DICOM DataSet.

25.184.3.5 bool gdcm::PDBHeader::LoadFromDataElement (DataElement const & *de*)

Load the PDB Header from a DataElement of a DataSet.

25.184.3.6 void gdcm::PDBHeader::Print (std::ostream & *os*) const

Print.

Referenced by gdcm::operator<<().

25.184.4 Friends And Related Function Documentation

25.184.4.1 std::ostream& operator<< (std::ostream & *_os*, const PDBHeader & *d*) [friend]

The documentation for this class was generated from the following file:

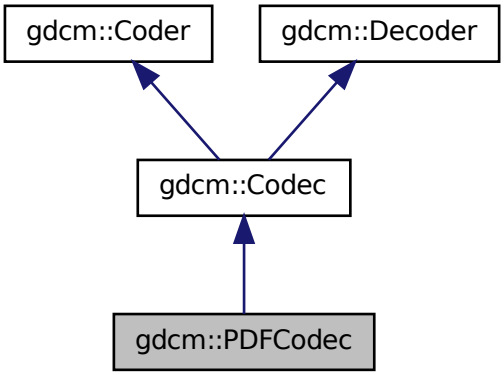
- gdcmPDBHeader.h

25.185 gdcm::PDFCodec Class Reference

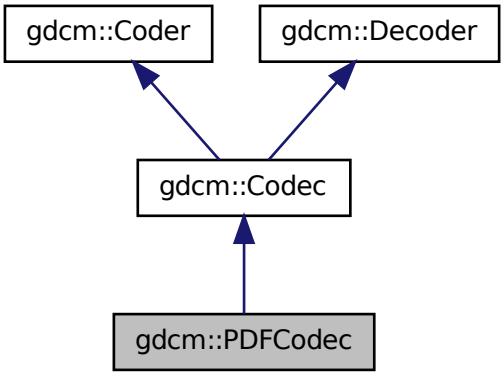
PDFCodec class.

`#include <gdcmPDFCodec.h>`

Inheritance diagram for gdcm::PDFCodec:



Collaboration diagram for gdcm::PDFCodec:



Public Member Functions

- PDFCodec ()
- ∼PDFCodec ()
- bool CanCode (TransferSyntax const &) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

25.185.1 Detailed Description

PDFCodec class.

25.185.2 Constructor & Destructor Documentation

25.185.2.1 gdcm::PDFCodec::PDFCodec ( )

25.185.2.2 gdcm::PDFCodec::∼PDFCodec ( )

25.185.3 Member Function Documentation

25.185.3.1 bool gdcm::PDFCodec::CanCode (TransferSyntax const &) const [inline],[virtual]

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

25.185.3.2 bool gdcm::PDFCodec::CanDecode (TransferSyntax const &) const [inline],[virtual]

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

25.185.3.3 bool gdcm::PDFCodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::Decoder.

The documentation for this class was generated from the following file:

- gdcmPDFCodec.h

25.186 gdcm::network::PDUFactory Class Reference

PDUFactory basically, given an initial byte, construct the appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

```
#include <gdcmPDUFactory.h>
```

Static Public Member Functions

- static BasePDU ∗ ConstructAbortPDU ()
- static BasePDU ∗ ConstructPDU (uint8_t itemtype)
- static BasePDU ∗ ConstructReleasePDU ()
- static std::vector< BasePDU ∗ > CreateCEchoPDU (const ULConnection &inConnection)
- static std::vector< BasePDU ∗ > CreateCFindPDU (const ULConnection &inConnection, const BaseRootQuery ∗inRootQuery)
- static std::vector< BasePDU ∗ > CreateCMovePDU (const ULConnection &inConnection, const BaseRootQuery ∗inRootQuery)
- static std::vector< BasePDU ∗ > CreateCStoreRQPDU (const ULConnection &inConnection, const File &file)
- static std::vector< BasePDU ∗ > CreateCStoreRSPPDU (const DataSet ∗inDataSet, const BasePDU ∗inPC)
- static EEventID DetermineEventByPDU (const BasePDU ∗inPDU)
- static std::vector < PresentationDataValue > GetPDVs (const std::vector< BasePDU ∗ > &inDataPDUs)

25.186.1 Detailed Description

PDUFactory basically, given an initial byte, construct the appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

25.186.2 Member Function Documentation

25.186.2.1 static BasePDU∗ gdcm::network::PDUFactory::ConstructAbortPDU ( ) [static]

25.186.2.2 static BasePDU∗ gdcm::network::PDUFactory::ConstructPDU (uint8_t *itemtype*) [static]

25.186.2.3 static **BasePDU**∗ gdcm::network::PDUFactory::ConstructReleasePDU ( ) [static]

25.186.2.4 static std::vector<**BasePDU**∗> gdcm::network::PDUFactory::CreateCEchoPDU (const **ULConnection** & *inConnection*) [static]

25.186.2.5 static std::vector<**BasePDU**∗> gdcm::network::PDUFactory::CreateCFindPDU (const **ULConnection** & *inConnection,* const **BaseRootQuery** ∗ *inRootQuery*) [static]

25.186.2.6 static std::vector<**BasePDU**∗> gdcm::network::PDUFactory::CreateCMovePDU (const **ULConnection** & *inConnection,* const **BaseRootQuery** ∗ *inRootQuery*) [static]

25.186.2.7 static std::vector<**BasePDU**∗> gdcm::network::PDUFactory::CreateCStoreRQPDU (const **ULConnection** & *inConnection,* const **File** & *file*) [static]

25.186.2.8 static std::vector<**BasePDU**∗> gdcm::network::PDUFactory::CreateCStoreRSPPDU (const **DataSet** ∗ *inDataSet,* const **BasePDU** ∗ *inPC*) [static]

25.186.2.9 static **EEventID** gdcm::network::PDUFactory::DetermineEventByPDU (const **BasePDU** ∗ *inPDU*) [static]

25.186.2.10 static std::vector<**PresentationDataValue**> gdcm::network::PDUFactory::GetPDVs (const std::vector< **BasePDU** ∗ > & *inDataPDUs*) [static]

The documentation for this class was generated from the following file:

- gdcmPDUFactory.h

25.187 gdcm::PersonName Class Reference

PersonName class.

`#include <gdcmPersonName.h>`

Public Member Functions

- unsigned int GetMaxLength () const
- unsigned int GetNumberOfComponents () const
- void Print (std::ostream &os) const
- void SetBlob (const std::vector< char > &v)
- void SetComponents (const char ∗comp1="", const char ∗comp2="", const char ∗comp3="", const char ∗comp4="", const char ∗comp5="")
- void SetComponents (const char ∗components[])

Public Attributes

- char Component [MaxNumberOfComponents][MaxLength+1]

Static Public Attributes

- static const unsigned int MaxLength = 64
- static const unsigned int MaxNumberOfComponents = 5

- static const char Padding = ' '
- static const char Separator = '^'

25.187.1 Detailed Description

PersonName class.

25.187.2 Member Function Documentation

25.187.2.1 unsigned int gdcm::PersonName::GetMaxLength ( ) const [inline]

25.187.2.2 unsigned int gdcm::PersonName::GetNumberOfComponents ( ) const [inline]

25.187.2.3 void gdcm::PersonName::Print (std::ostream & *os*) const [inline]

25.187.2.4 void gdcm::PersonName::SetBlob (const std::vector< char > & *v*) [inline]

25.187.2.5 void gdcm::PersonName::SetComponents (const char * *comp1* = " ", const char * *comp2* = " ", const char * *comp3* = " ", const char * *comp4* = " ", const char * *comp5* = " ") [inline]

25.187.2.6 void gdcm::PersonName::SetComponents (const char * *components[]*) [inline]

25.187.3 Member Data Documentation

25.187.3.1 char gdcm::PersonName::Component[MaxNumberOfComponents][MaxLength+1]

25.187.3.2 const unsigned int gdcm::PersonName::MaxLength = 64 [static]

25.187.3.3 const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5 [static]

25.187.3.4 const char gdcm::PersonName::Padding = ' ' [static]

25.187.3.5 const char gdcm::PersonName::Separator = '^' [static]

The documentation for this class was generated from the following file:

- gdcmPersonName.h

25.188 gdcm::PhotometricInterpretation Class Reference

Class to represent an PhotometricInterpretation.

```
#include <gdcmPhotometricInterpretation.h>
```

Public Types

- enum PIType {
  UNKNOW = 0,
  MONOCHROME1,
  MONOCHROME2,
  PALETTE_COLOR,
  RGB,
  HSV,
  ARGB,
  CMYK,
  YBR_FULL,
  YBR_FULL_422,
  YBR_PARTIAL_422,
  YBR_PARTIAL_420,
  YBR_ICT,
  YBR_RCT,
  PI_END }

Public Member Functions

- PhotometricInterpretation (PIType pi=UNKNOW)
- unsigned short GetSamplesPerPixel () const

  return the value for Sample Per Pixel associated with a particular Photometric Interpretation
- const char ∗ GetString () const
- PIType GetType () const
- bool IsLossless () const
- bool IsLossy () const
- bool IsSameColorSpace (PhotometricInterpretation const &pi) const
- operator PIType () const

Static Public Member Functions

- static const char ∗ GetPIString (PIType pi)
- static PIType GetPIType (const char ∗pi)
- static bool IsRetired (PIType pi)

Friends

- std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &pi)

25.188.1 Detailed Description

Class to represent an PhotometricInterpretation.

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, HelloVizWorld.cxx, and iU22tomultisc.cxx.

25.188.2 Member Enumeration Documentation

25.188.2.1 enum gdcm::PhotometricInterpretation::PIType

Enumerator:

UNKNOW
MONOCHROME1
MONOCHROME2
PALETTE_COLOR
RGB
HSV
ARGB
CMYK
YBR_FULL
YBR_FULL_422
YBR_PARTIAL_422
YBR_PARTIAL_420
YBR_ICT
YBR_RCT
PI_END

25.188.3 Constructor & Destructor Documentation

25.188.3.1 gdcm::PhotometricInterpretation::PhotometricInterpretation (PIType *pi* = UNKNOW) [inline]

25.188.4 Member Function Documentation

25.188.4.1 static const char* gdcm::PhotometricInterpretation::GetPIString (PIType *pi*) [static]

Referenced by gdcm::operator<<().

25.188.4.2 static PIType gdcm::PhotometricInterpretation::GetPIType (const char * *pi*) [static]

25.188.4.3 unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel ( ) const

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

25.188.4.4 const char* gdcm::PhotometricInterpretation::GetString ( ) const

25.188.4.5 PIType gdcm::PhotometricInterpretation::GetType ( ) const [inline]

25.188.4.6 bool gdcm::PhotometricInterpretation::IsLossless ( ) const

25.188.4.7 bool gdcm::PhotometricInterpretation::IsLossy ( ) const

25.188.4.8 static bool gdcm::PhotometricInterpretation::IsRetired (PIType *pi*) [static]

25.188.4.9 bool gdcm::PhotometricInterpretation::IsSameColorSpace (PhotometricInterpretation const & *pi*) const

25.188.4.10 gdcm::PhotometricInterpretation::operator PIType ( ) const [inline]

25.188.5 Friends And Related Function Documentation

25.188.5.1 std::ostream& operator<< (std::ostream & *os,* const PhotometricInterpretation & *pi*) [friend]

The documentation for this class was generated from the following file:

- gdcmPhotometricInterpretation.h

25.189 gdcm::PixelFormat Class Reference

PixelFormat.

```
#include <gdcmPixelFormat.h>
```

Public Types

- enum ScalarType {
  UINT8,
  INT8,
  UINT12,
  INT12,
  UINT16,
  INT16,
  UINT32,
  INT32,
  FLOAT16,
  FLOAT32,
  FLOAT64,
  SINGLEBIT,
  UNKNOWN }

Public Member Functions

- PixelFormat (unsigned short samplesperpixel=1, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0)
- PixelFormat (ScalarType st)
- ~PixelFormat ()
- unsigned short GetBitsAllocated () const

  BitsAllocated see Tag (0028,0100) US Bits Allocated.
- unsigned short GetBitsStored () const

  BitsStored see Tag (0028,0101) US Bits Stored.
- unsigned short GetHighBit () const

  HighBit see Tag (0028,0102) US High Bit.
- int64_t GetMax () const

  return the max possible of the pixel

- int64_t GetMin () const

  return the min possible of the pixel
- unsigned short GetPixelRepresentation () const

  PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.
- uint8_t GetPixelSize () const
- unsigned short GetSamplesPerPixel () const
- ScalarType GetScalarType () const

  ScalarType does not take into account the sample per pixel.
- const char * GetScalarTypeAsString () const
- bool IsValid ()

  return IsValid
- operator ScalarType () const
- bool operator!= (ScalarType st) const
- bool operator!= (const PixelFormat &pf) const
- bool operator== (ScalarType st) const
- bool operator== (const PixelFormat &pf) const
- void Print (std::ostream &os) const

  Print.
- void SetBitsAllocated (unsigned short ba)
- void SetBitsStored (unsigned short bs)
- void SetHighBit (unsigned short hb)
- void SetPixelRepresentation (unsigned short pr)
- void SetSamplesPerPixel (unsigned short spp)
- void SetScalarType (ScalarType st)

Protected Member Functions

- bool Validate ()

  When image with 24/24/23 was read, need to validate.

Friends

- class Bitmap
- std::ostream & operator<< (std::ostream &_os, const PixelFormat &pf)

25.189.1 Detailed Description

PixelFormat.

Note

By default the Pixel Type will be instanciated with the following parameters:

- SamplesPerPixel : 1
- BitsAllocated : 8
- BitsStored : 8
- HighBit : 7
- PixelRepresentation : 0

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, iU22tomultisc.cxx, and threadgdcm.cxx.

25.189.2 Member Enumeration Documentation

25.189.2.1 enum gdcm::PixelFormat::ScalarType

Enumerator:

UINT8
INT8
UINT12
INT12
UINT16
INT16
UINT32
INT32
FLOAT16
FLOAT32
FLOAT64
SINGLEBIT
UNKNOWN

25.189.3 Constructor & Destructor Documentation

25.189.3.1 gdcm::PixelFormat::PixelFormat (unsigned short *samplesperpixel* = 1, unsigned short *bitsallocated* = 8, unsigned short *bitsstored* = 8, unsigned short *highbit* = 7, unsigned short *pixelrepresentation* = 0) [inline], [explicit]

25.189.3.2 gdcm::PixelFormat::PixelFormat (ScalarType *st*)

25.189.3.3 gdcm::PixelFormat::~PixelFormat ( ) [inline]

25.189.4 Member Function Documentation

25.189.4.1 unsigned short gdcm::PixelFormat::GetBitsAllocated ( ) const [inline]

BitsAllocated see Tag (0028,0100) US Bits Allocated.

Examples:

GetJPEGSamplePrecision.cxx.

25.189.4.2 unsigned short gdcm::PixelFormat::GetBitsStored ( ) const [inline]

BitsStored see Tag (0028,0101) US Bits Stored.

Examples:

GetJPEGSamplePrecision.cxx.

25.189.4.3 unsigned short gdcm::PixelFormat::GetHighBit ( ) const [inline]

HighBit see Tag (0028,0102) US High Bit.

25.189.4.4 int64_t gdcm::PixelFormat::GetMax ( ) const

return the max possible of the pixel

25.189.4.5 int64_t gdcm::PixelFormat::GetMin ( ) const

return the min possible of the pixel

25.189.4.6 unsigned short gdcm::PixelFormat::GetPixelRepresentation ( ) const [inline]

PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.

25.189.4.7 uint8_t gdcm::PixelFormat::GetPixelSize ( ) const

return the size of the pixel This is the number of words it would take to store one pixel

Warning

the return value takes into account the SamplesPerPixel
in the rare case when BitsAllocated == 12, the function assume word padding and value returned will be identical as if BitsAllocated == 16

Examples:

threadgdcm.cxx.

25.189.4.8 unsigned short gdcm::PixelFormat::GetSamplesPerPixel ( ) const

Samples Per Pixel see (0028,0002) US Samples Per Pixel DICOM - only allows 1, 3 and 4 as valid value. Other value are undefined behavior.

Examples:

threadgdcm.cxx.

25.189.4.9 ScalarType gdcm::PixelFormat::GetScalarType ( ) const

ScalarType does not take into account the sample per pixel.

25.189.4.10 const char* gdcm::PixelFormat::GetScalarTypeAsString ( ) const

25.189.4.11 bool gdcm::PixelFormat::IsValid ( )

return IsValid

25.189.4.12 gdcm::PixelFormat::operator ScalarType ( ) const [inline]

25.189.4.13 bool gdcm::PixelFormat::operator!= (ScalarType *st*) const [inline]

25.189.4.14 bool gdcm::PixelFormat::operator!= (const PixelFormat & *pf*) const [inline]

25.189.4.15 bool gdcm::PixelFormat::operator== (ScalarType *st*) const [inline]

25.189.4.16 bool gdcm::PixelFormat::operator== (const PixelFormat & *pf*) const [inline]

25.189.4.17 void gdcm::PixelFormat::Print (std::ostream & *os*) const

Print.

Referenced by gdcm::operator<<().

25.189.4.18 void gdcm::PixelFormat::SetBitsAllocated (unsigned short *ba*) [inline]

25.189.4.19 void gdcm::PixelFormat::SetBitsStored (unsigned short *bs*) [inline]

25.189.4.20 void gdcm::PixelFormat::SetHighBit (unsigned short *hb*) [inline]

25.189.4.21 void gdcm::PixelFormat::SetPixelRepresentation (unsigned short *pr*) [inline]

25.189.4.22 void gdcm::PixelFormat::SetSamplesPerPixel (unsigned short *spp*) [inline]

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, and GenFakeImage.cxx.

References gdcmAssertMacro.

25.189.4.23 void gdcm::PixelFormat::SetScalarType (ScalarType *st*)

Set PixelFormat based only on the ScalarType

Warning

: You need to call SetScalarType *before* SetSamplesPerPixel

25.189.4.24 bool gdcm::PixelFormat::Validate ( ) [protected]

When image with 24/24/23 was read, need to validate.

Referenced by gdcm::Bitmap::SetPixelFormat().

25.189.5 Friends And Related Function Documentation

25.189.5.1 friend class Bitmap [friend]

25.189.5.2 std::ostream& operator<< (std::ostream & _os, const PixelFormat & pf) [friend]

The documentation for this class was generated from the following file:

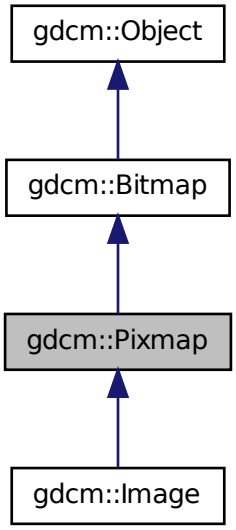
- gdcmPixelFormat.h

25.190 gdcm::Pixmap Class Reference

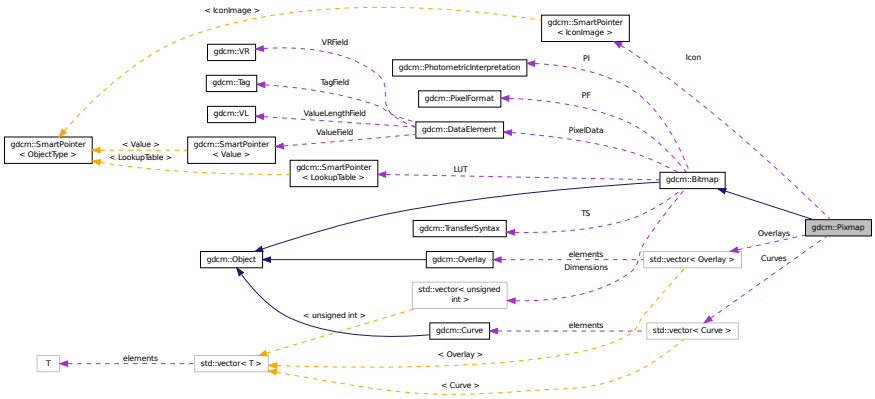
Pixmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

```
#include <gdcmPixmap.h>
```

Inheritance diagram for gdcm::Pixmap:



Collaboration diagram for gdcm::Pixmap:



Public Member Functions

- Pixmap ()
- ~Pixmap ()
- bool AreOverlaysInPixelData () const

  returns if Overlays are stored in the unused bit of the pixel data:
- Curve & GetCurve (unsigned int i=0)

  Curve: group 50xx.
- const Curve & GetCurve (unsigned int i=0) const
- const IconImage & GetIconImage () const

  Set/Get Icon Image.
- IconImage & GetIconImage ()
- size_t GetNumberOfCurves () const
- size_t GetNumberOfOverlays () const
- Overlay & GetOverlay (size_t i=0)

  Overlay: group 60xx.
- const Overlay & GetOverlay (size_t i=0) const
- void Print (std::ostream &) const
- void RemoveOverlay (size_t i)
- void SetIconImage (IconImage const &ii)
- void SetNumberOfCurves (size_t n)
- void SetNumberOfOverlays (size_t n)

Protected Attributes

- std::vector< Curve > Curves
- SmartPointer< IconImage > Icon
- std::vector< Overlay > Overlays

Additional Inherited Members

25.190.1 Detailed Description

Pixmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

See also

PixmapReader

25.190.2 Constructor & Destructor Documentation

25.190.2.1 gdcm::Pixmap::Pixmap ( )

25.190.2.2 gdcm::Pixmap::∼Pixmap ( )

25.190.3 Member Function Documentation

25.190.3.1 bool gdcm::Pixmap::AreOverlaysInPixelData ( ) const [virtual]

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from gdcm::Bitmap.

25.190.3.2 Curve& gdcm::Pixmap::GetCurve (unsigned int *i* = 0) [inline]

Curve: group 50xx.

25.190.3.3 const Curve& gdcm::Pixmap::GetCurve (unsigned int *i* = 0) const [inline]

25.190.3.4 const IconImage& gdcm::Pixmap::GetIconImage ( ) const [inline]

Set/Get Icon Image.

25.190.3.5 IconImage& gdcm::Pixmap::GetIconImage ( ) [inline]

25.190.3.6 size_t gdcm::Pixmap::GetNumberOfCurves ( ) const [inline]

25.190.3.7 size_t gdcm::Pixmap::GetNumberOfOverlays ( ) const [inline]

25.190.3.8 Overlay& gdcm::Pixmap::GetOverlay (size_t *i* = 0) [inline]

Overlay: group 60xx.

25.190.3.9 const Overlay& gdcm::Pixmap::GetOverlay (size_t *i* = 0) const [inline]

25.190.3.10 void gdcm::Pixmap::Print (std::ostream &) const [virtual]

Reimplemented from gdcm::Bitmap.

Reimplemented in gdcm::Image.

25.190.3.11 void gdcm::Pixmap::RemoveOverlay (size_t *i*) [inline]

25.190.3.12 void gdcm::Pixmap::SetIconImage (IconImage const & *ii*) [inline]

25.190.3.13 void gdcm::Pixmap::SetNumberOfCurves (size_t *n*) [inline]

25.190.3.14 void gdcm::Pixmap::SetNumberOfOverlays (size_t *n*) [inline]

25.190.4 Member Data Documentation

25.190.4.1 std::vector<Curve> gdcm::Pixmap::Curves [protected]

25.190.4.2 SmartPointer<IconImage> gdcm::Pixmap::Icon [protected]

25.190.4.3 std::vector<Overlay> gdcm::Pixmap::Overlays [protected]

The documentation for this class was generated from the following file:

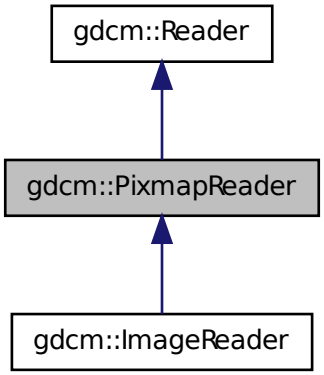
- gdcmPixmap.h

25.191 gdcm::PixmapReader Class Reference

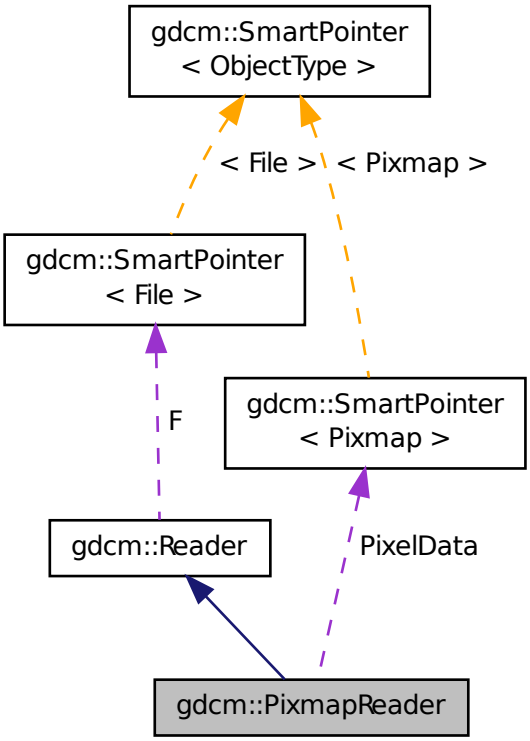
PixmapReader.

`#include <gdcmPixmapReader.h>`

Inheritance diagram for gdcm::PixmapReader:



Collaboration diagram for gdcm::PixmapReader:



Public Member Functions

- PixmapReader ()
- virtual ~PixmapReader ()
- const Pixmap & GetPixmap () const

  Return the read image (need to call Read() first)
- Pixmap & GetPixmap ()
- virtual bool Read ()

Protected Member Functions

- virtual bool ReadACRNEMAImage ()
- virtual bool ReadImage (MediaStorage const &ms)

Protected Attributes

- SmartPointer< Pixmap > PixelData

25.191.1 Detailed Description

PixmapReader.

Note

its role is to convert the DICOM DataSet into a gdcm::Pixmap representation By default it is also loading the lookup table and overlay when found as they impact the rendering or the image

See PS 3.3-2008, Table C.7-11b IMAGE PIXEL MACRO ATTRIBUTES for the list of attribute that belong to what gdcm calls a 'Pixmap'

Warning

the API ReadUpToTag and ReadSelectedTag

See also

Pixmap

25.191.2 Constructor & Destructor Documentation

25.191.2.1 gdcm::PixmapReader::PixmapReader ( )

25.191.2.2 virtual gdcm::PixmapReader::~PixmapReader ( ) [virtual]

25.191.3 Member Function Documentation

25.191.3.1 const Pixmap& gdcm::PixmapReader::GetPixmap ( ) const

Return the read image (need to call Read() first)

25.191.3.2 Pixmap& gdcm::PixmapReader::GetPixmap ( )

25.191.3.3 virtual bool gdcm::PixmapReader::Read ( ) [virtual]

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an Pixmap.

Reimplemented from gdcm::Reader.

Reimplemented in gdcm::ImageReader.

25.191.3.4 virtual bool gdcm::PixmapReader::ReadACRNEMAImage ( ) [protected],[virtual]

Reimplemented in gdcm::ImageReader.

25.191.3.5 virtual bool gdcm::PixmapReader::ReadImage (MediaStorage const & *ms*) [protected],[virtual]

Reimplemented in gdcm::ImageReader.

25.191.4 Member Data Documentation

25.191.4.1 SmartPointer<Pixmap> gdcm::PixmapReader::PixelData [protected]

The documentation for this class was generated from the following file:

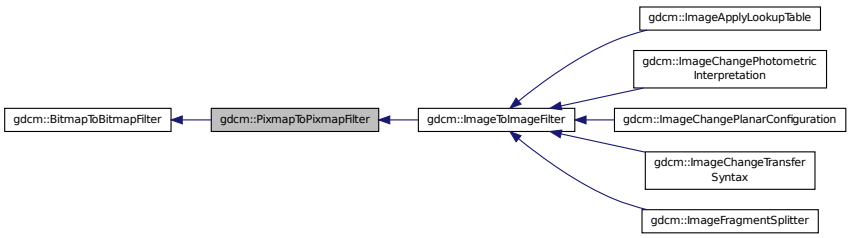
- gdcmPixmapReader.h

25.192 gdcm::PixmapToPixmapFilter Class Reference

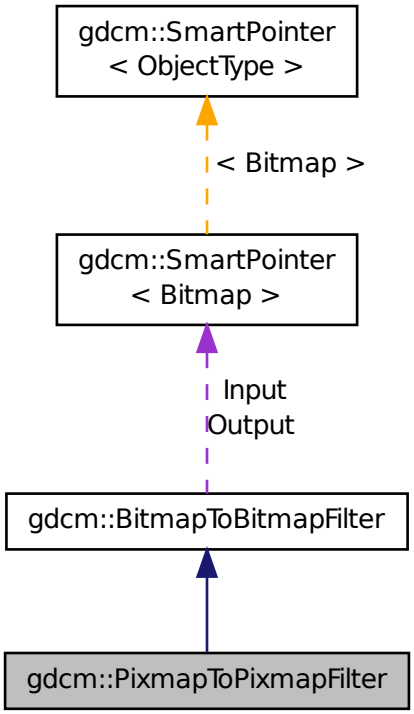
PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.

`#include <gdcmPixmapToPixmapFilter.h>`

Inheritance diagram for gdcm::PixmapToPixmapFilter:



Collaboration diagram for gdcm::PixmapToPixmapFilter:



Public Member Functions

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.

Additional Inherited Members

25.192.1 Detailed Description

PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.

25.192.2 Constructor & Destructor Documentation

25.192.2.1 gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ( )

25.192.2.2 gdcm::PixmapToPixmapFilter::~PixmapToPixmapFilter ( ) [inline]

25.192.3 Member Function Documentation

25.192.3.1 Pixmap& gdcm::PixmapToPixmapFilter::GetInput ( )

Reimplemented in gdcm::ImageToImageFilter.

25.192.3.2 const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput ( ) const

Get Output image.

Reimplemented from gdcm::BitmapToBitmapFilter.

Reimplemented in gdcm::ImageToImageFilter.

The documentation for this class was generated from the following file:

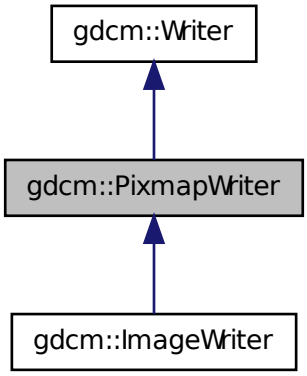
- gdcmPixmapToPixmapFilter.h

25.193 gdcm::PixmapWriter Class Reference

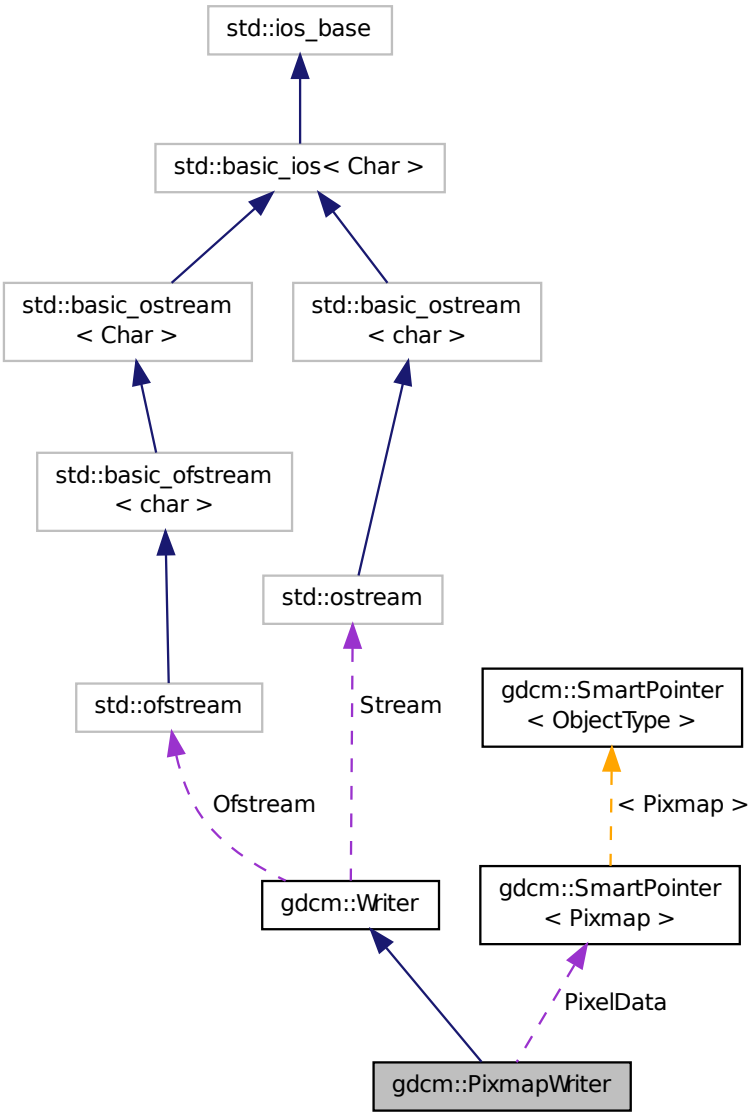
PixmapWriter This class will takes two inputs:

`#include <gdcmPixmapWriter.h>`

Inheritance diagram for gdcm::PixmapWriter:



Collaboration diagram for gdcm::PixmapWriter:



Public Member Functions

- PixmapWriter ()
- ~PixmapWriter ()
- virtual const Pixmap & GetImage () const
- virtual Pixmap & GetImage ()
- const Pixmap & GetPixmap () const
- Pixmap & GetPixmap ()
- virtual void SetImage (Pixmap const &img)

- void SetPixmap (Pixmap const &img)
- bool Write ()

  Write.

Protected Member Functions

- void DoIconImage (DataSet &ds, Pixmap const &image)
- bool PrepareWrite ()

Protected Attributes

- SmartPointer< Pixmap > PixelData

25.193.1 Detailed Description

PixmapWriter This class will takes two inputs:

1. The DICOM DataSet
2. The Image input It will override any info from the Image over the DataSet.

For instance when one read in a lossy compressed image and write out as unencapsulated (ie implicitely lossless) then some attribute are definitely needed to mark this dataset as Lossy (typically 0028,2114)

25.193.2 Constructor & Destructor Documentation

25.193.2.1 gdcm::PixmapWriter::PixmapWriter ( )

25.193.2.2 gdcm::PixmapWriter::~PixmapWriter ( )

25.193.3 Member Function Documentation

25.193.3.1 void gdcm::PixmapWriter::DoIconImage (DataSet & *ds*, Pixmap const & *image*) [protected]

25.193.3.2 virtual const Pixmap& gdcm::PixmapWriter::GetImage ( ) const [inline],[virtual]

Set/Get Pixmap to be written It will overwrite anything Pixmap infos found in DataSet (see parent class to see how to pass dataset)

Reimplemented in gdcm::ImageWriter.

25.193.3.3 virtual Pixmap& gdcm::PixmapWriter::GetImage ( ) [inline],[virtual]

Reimplemented in gdcm::ImageWriter.

25.193.3.4 const Pixmap& gdcm::PixmapWriter::GetPixmap ( ) const [inline]

25.193.3.5 Pixmap& gdcm::PixmapWriter::GetPixmap ( ) [inline]

25.193.3.6 bool gdcm::PixmapWriter::PrepareWrite ( ) [protected]

25.193.3.7 virtual void gdcm::PixmapWriter::SetImage (Pixmap const & *img*) [virtual]

Examples:

CompressImage.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, and MergeTwoFiles.cxx.

25.193.3.8 void gdcm::PixmapWriter::SetPixmap (Pixmap const & *img*)

25.193.3.9 bool gdcm::PixmapWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::Writer.

Reimplemented in gdcm::ImageWriter.

25.193.4 Member Data Documentation

25.193.4.1 SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData [protected]

The documentation for this class was generated from the following file:

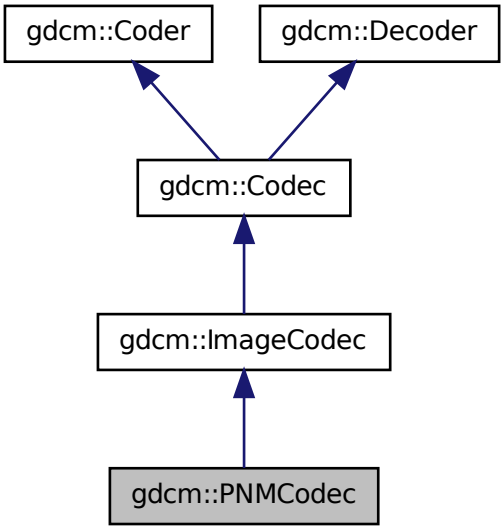
- gdcmPixmapWriter.h

25.194 gdcm::PNMCodec Class Reference

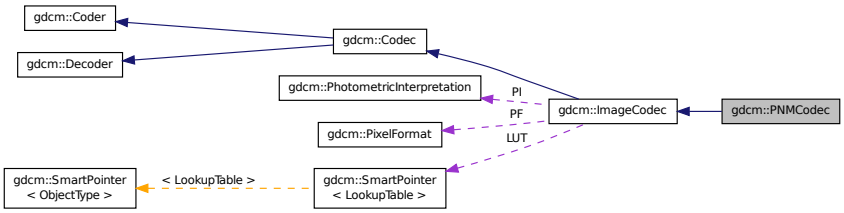
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/`.

`#include <gdcmPNMCodec.h>`

Inheritance diagram for gdcm::PNMCodec:



Collaboration diagram for gdcm::PNMCodec:



Public Member Functions

- PNMCodec ()
- ∼PNMCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool Read (const char ∗filename, DataElement &out) const
- void SetBufferLength (unsigned long l)

- bool Write (const char *filename, const DataElement &out) const

Additional Inherited Members

25.194.1 Detailed Description

Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/`.

Note

Only support P5 & P6 PNM file (binary grayscale and binary rgb)

Examples:

ExtractIconFromFile.cxx.

25.194.2 Constructor & Destructor Documentation

25.194.2.1 gdcm::PNMCodec::PNMCodec ( )

25.194.2.2 gdcm::PNMCodec::~PNMCodec ( )

25.194.3 Member Function Documentation

25.194.3.1 bool gdcm::PNMCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.194.3.2 bool gdcm::PNMCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.194.3.3 unsigned long gdcm::PNMCodec::GetBufferLength ( ) const [inline]

25.194.3.4 bool gdcm::PNMCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

25.194.3.5 bool gdcm::PNMCodec::Read (const char * *filename*, DataElement & *out*) const

25.194.3.6 void gdcm::PNMCodec::SetBufferLength (unsigned long *l*) [inline]

25.194.3.7 bool gdcm::PNMCodec::Write (const char * *filename*, const DataElement & *out*) const

Examples:

ExtractIconFromFile.cxx.

The documentation for this class was generated from the following file:

- gdcmPNMCodec.h

25.195 gdcm::Preamble Class Reference

DICOM Preamble (Part 10)

`#include <gdcmPreamble.h>`

Public Member Functions

- Preamble ()
- Preamble (Preamble const &)
- ∼Preamble ()
- void Clear ()
- void Create ()
- const char ∗ GetInternal () const
- VL GetLength () const
- bool IsEmpty () const
- Preamble & operator= (Preamble const &)
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void Remove ()
- void Valid ()
- std::ostream const & Write (std::ostream &os) const

Protected Member Functions

- bool IsValid () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Preamble &_val)

25.195.1 Detailed Description

DICOM Preamble (Part 10)

25.195.2 Constructor & Destructor Documentation

25.195.2.1 gdcm::Preamble::Preamble ( )

25.195.2.2 gdcm::Preamble::∼Preamble ( )

25.195.2.3 gdcm::Preamble::Preamble (Preamble const &) [inline]

25.195.3 Member Function Documentation

25.195.3.1 void gdcm::Preamble::Clear ( )

25.195.3.2 void gdcm::Preamble::Create ( )

25.195.3.3 const char* gdcm::Preamble::GetInternal ( ) const [inline]

25.195.3.4 VL gdcm::Preamble::GetLength ( ) const [inline]

25.195.3.5 bool gdcm::Preamble::IsEmpty ( ) const [inline]

25.195.3.6 bool gdcm::Preamble::IsValid ( ) const [inline], [protected]

25.195.3.7 Preamble& gdcm::Preamble::operator= (Preamble const &) [inline]

25.195.3.8 void gdcm::Preamble::Print (std::ostream & *os*) const

25.195.3.9 std::istream& gdcm::Preamble::Read (std::istream & *is*)

25.195.3.10 void gdcm::Preamble::Remove ( )

25.195.3.11 void gdcm::Preamble::Valid ( )

25.195.3.12 std::ostream const& gdcm::Preamble::Write (std::ostream & *os*) const

25.195.4 Friends And Related Function Documentation

25.195.4.1 std::ostream& operator<< (std::ostream & *_os,* const Preamble & *_val*) [friend]

The documentation for this class was generated from the following file:

- gdcmPreamble.h

25.196 gdcm::PresentationContext Class Reference

PresentationContext.

```
#include <gdcmPresentationContext.h>
```

Public Types

- typedef TransferSyntaxArrayType::size_type SizeType
- typedef std::vector< std::string > TransferSyntaxArrayType

Public Member Functions

- PresentationContext ()

- PresentationContext (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
- void AddTransferSyntax (const char *tsstr)
- const char * GetAbstractSyntax () const
- SizeType GetNumberOfTransferSyntaxes () const
- uint8_t GetPresentationContextID () const
- const char * GetTransferSyntax (SizeType i) const
- bool operator== (const PresentationContext &pc) const
- void Print (std::ostream &os) const
- void SetAbstractSyntax (const char *as)
- void SetPresentationContextID (uint8_t id)

25.196.1 Detailed Description

PresentationContext.

See also

PresentationContextAC PresentationContextRQ

25.196.2 Member Typedef Documentation

25.196.2.1 typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType

25.196.2.2 typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType

25.196.3 Constructor & Destructor Documentation

25.196.3.1 gdcm::PresentationContext::PresentationContext ( )

25.196.3.2 gdcm::PresentationContext::PresentationContext (UIDs::TSName *asname,* UIDs::TSName *tsname =* UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)

Initialize Presentation Context with AbstractSyntax set to asname and with a single TransferSyntax set to tsname (dfault to Implicit VR LittleEndian when not specified).

25.196.4 Member Function Documentation

25.196.4.1 void gdcm::PresentationContext::AddTransferSyntax (const char * *tsstr*)

25.196.4.2 const char* gdcm::PresentationContext::GetAbstractSyntax ( ) const [inline]

25.196.4.3 SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes ( ) const [inline]

25.196.4.4 uint8_t gdcm::PresentationContext::GetPresentationContextID ( ) const

25.196.4.5 const char* gdcm::PresentationContext::GetTransferSyntax (SizeType *i*) const [inline]

25.196.4.6 bool gdcm::PresentationContext::operator== (const PresentationContext & *pc*) const [inline]

25.196.4.7 void gdcm::PresentationContext::Print (std::ostream & *os*) const

25.196.4.8 void gdcm::PresentationContext::SetAbstractSyntax (const char * *as*) [inline]

25.196.4.9 void gdcm::PresentationContext::SetPresentationContextID (uint8_t *id*)

The documentation for this class was generated from the following file:

- gdcmPresentationContext.h

25.197 gdcm::network::PresentationContextAC Class Reference

PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

```
#include <gdcmPresentationContextAC.h>
```

Public Member Functions

- PresentationContextAC ()
- uint8_t GetPresentationContextID () const
- TransferSyntaxSub const & GetTransferSyntax () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetPresentationContextID (uint8_t id)
- void SetTransferSyntax (TransferSyntaxSub const &ts)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.197.1 Detailed Description

PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

See also

PresentationContext

25.197.2 Constructor & Destructor Documentation

25.197.2.1 gdcm::network::PresentationContextAC::PresentationContextAC ( )

25.197.3 Member Function Documentation

25.197.3.1 uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID ( ) const [inline]

25.197.3.2 TransferSyntaxSub const& gdcm::network::PresentationContextAC::GetTransferSyntax ( ) const [inline]

25.197.3.3 void gdcm::network::PresentationContextAC::Print (std::ostream & *os*) const

25.197.3.4 std::istream& gdcm::network::PresentationContextAC::Read (std::istream & *is*)

25.197.3.5 void gdcm::network::PresentationContextAC::SetPresentationContextID (uint8_t *id*)

25.197.3.6 void gdcm::network::PresentationContextAC::SetTransferSyntax (TransferSyntaxSub const & *ts*)

25.197.3.7 size_t gdcm::network::PresentationContextAC::Size ( ) const

25.197.3.8 const std::ostream& gdcm::network::PresentationContextAC::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

- gdcmPresentationContextAC.h

25.198 gdcm::PresentationContextGenerator Class Reference

PresentationContextGenerator This class is responsible for generating the proper PresentationContext that will be used in subsequent operation during a DICOM Query/Retrieve association. The step of the association is very sensible as special care need to be taken to explicitly define what instance are going to be send and how they are encoded.

```
#include <gdcmPresentationContextGenerator.h>
```

Public Types

- typedef std::vector
  < PresentationContext > PresentationContextArrayType
- typedef
  PresentationContextArrayType::size_type SizeType

Public Member Functions

- PresentationContextGenerator ()
- bool GenerateFromFilenames (const Directory::FilenamesType &files)
- bool GenerateFromUID (UIDs::TSName asname)

  Generate the PresentationContext array from a UID (eg. VerificationSOPClass)
- PresentationContextArrayType
  const & GetPresentationContexts ()
- void SetDefaultTransferSyntax (const TransferSyntax &ts)

  Not implemented for now. GDCM internally uses Implicit Little Endian.
- void SetMergeModeToAbstractSyntax ()
- void SetMergeModeToTransferSyntax ()

Protected Member Functions

- bool AddPresentationContext (const char ∗as, const char ∗ts)
- const char ∗ GetDefaultTransferSyntax () const

25.198.1 Detailed Description

PresentationContextGenerator This class is responsible for generating the proper PresentationContext that will be used in subsequent operation during a DICOM Query/Retrieve association. The step of the association is very sensible as special care need to be taken to explicitly define what instance are going to be send and how they are encoded.

For example a PresentationContext will express that negotiation requires that CT Image Storage are send using JPEG Lossless, while US Image Storage are sent using RLE Transfer Syntax.

Two very different API are exposed one which will always default to little endian transfer syntax see GenerateFromUID() This API is used for C-ECHO, C-FIND and C-MOVE (SCU). Another API: GenerateFromFilenames() is used for C-STORE (SCU) as it will loop over all filenames argument to detect the actual encoding. and therefore find the proper encoding to be used.

Two modes are available. The default mode (SetMergeModeToAbstractSyntax) append PresentationContext (one AbstractSyntax and one TransferSyntax), as long a they are different. Eg MR Image Storage/JPEG2000 and MR Image Storage/JPEGLossless would be considered different. the other mode SetMergeModeToTransferSyntax merge any new TransferSyntax to the already existing PresentationContext in order to re-use the same AbstractSyntax.

See also

PresentationContext

Examples:

CStoreQtProgress.cxx.

25.198.2 Member Typedef Documentation

25.198.2.1 typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType

25.198.2.2 typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType

25.198.3 Constructor & Destructor Documentation

25.198.3.1 gdcm::PresentationContextGenerator::PresentationContextGenerator ( )

25.198.4 Member Function Documentation

25.198.4.1 bool gdcm::PresentationContextGenerator::AddPresentationContext (const char * *as*, const char * *ts*) [protected]

25.198.4.2 bool gdcm::PresentationContextGenerator::GenerateFromFilenames (const Directory::FilenamesType & *files*)

Generate the PresentationContext array from a File-Set. File specified needs to be valid DICOM files. Used for C-STORE operations

Examples:

CStoreQtProgress.cxx.

25.198.4.3 bool gdcm::PresentationContextGenerator::GenerateFromUID (UIDs::TSName *asname*)

Generate the PresentationContext array from a UID (eg. VerificationSOPClass)

25.198.4.4 const char* gdcm::PresentationContextGenerator::GetDefaultTransferSyntax ( ) const [protected]

25.198.4.5 PresentationContextArrayType const& gdcm::PresentationContextGenerator::GetPresentationContexts ( ) [inline]

Examples:

CStoreQtProgress.cxx.

25.198.4.6 void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (const TransferSyntax & *ts*)

Not implemented for now. GDCM internally uses Implicit Little Endian.

25.198.4.7 void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ( )

25.198.4.8 void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ( )

The documentation for this class was generated from the following file:

- gdcmPresentationContextGenerator.h

25.199 gdcm::network::PresentationContextRQ Class Reference

PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

```
#include <gdcmPresentationContextRQ.h>
```

Public Types

- typedef std::vector
  < TransferSyntaxSub >
  ::size_type SizeType

Public Member Functions

- PresentationContextRQ ()
- PresentationContextRQ (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
- PresentationContextRQ (const PresentationContext &pc)
- void AddTransferSyntax (TransferSyntaxSub const &ts)
- AbstractSyntax const & GetAbstractSyntax () const
- AbstractSyntax & GetAbstractSyntax ()
- SizeType GetNumberOfTransferSyntaxes () const
- uint8_t GetPresentationContextID () const
- TransferSyntaxSub const & GetTransferSyntax (SizeType i) const
- TransferSyntaxSub & GetTransferSyntax (SizeType i)
- std::vector< TransferSyntaxSub >
  const & GetTransferSyntaxes () const

- bool operator== (const PresentationContextRQ &pc) const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetAbstractSyntax (AbstractSyntax const &as)
- void SetPresentationContextID (uint8_t id)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.199.1 Detailed Description

PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

See also

PresentationContextAC

25.199.2 Member Typedef Documentation

25.199.2.1 typedef std::vector<TransferSyntaxSub>::size_type gdcm::network::PresentationContextRQ::SizeType

25.199.3 Constructor & Destructor Documentation

25.199.3.1 gdcm::network::PresentationContextRQ::PresentationContextRQ ( )

25.199.3.2 gdcm::network::PresentationContextRQ::PresentationContextRQ (UIDs::TSName *asname,* UIDs::TSName *tsname* = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)

Initialize Presentation Context with AbstractSyntax set to asname and with a single TransferSyntax set to tsname (dfault to Implicit VR LittleEndian when not specified).

25.199.3.3 gdcm::network::PresentationContextRQ::PresentationContextRQ (const PresentationContext & *pc*)

25.199.4 Member Function Documentation

25.199.4.1 void gdcm::network::PresentationContextRQ::AddTransferSyntax (TransferSyntaxSub const & *ts*)

25.199.4.2 AbstractSyntax const& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) const [inline]

25.199.4.3 AbstractSyntax& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) [inline]

25.199.4.4 SizeType gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes ( ) const [inline]

25.199.4.5 uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID ( ) const

25.199.4.6 TransferSyntaxSub const& gdcm::network::PresentationContextRQ::GetTransferSyntax (SizeType *i*) const [inline]

25.199.4.7 TransferSyntaxSub& gdcm::network::PresentationContextRQ::GetTransferSyntax (SizeType *i*) [inline]

25.199.4.8 std::vector<TransferSyntaxSub> const& gdcm::network::PresentationContextRQ::GetTransferSyntaxes ( ) const [inline]

25.199.4.9 bool gdcm::network::PresentationContextRQ::operator== (const PresentationContextRQ & *pc*) const [inline]

25.199.4.10 void gdcm::network::PresentationContextRQ::Print (std::ostream & *os*) const

25.199.4.11 std::istream& gdcm::network::PresentationContextRQ::Read (std::istream & *is*)

25.199.4.12 void gdcm::network::PresentationContextRQ::SetAbstractSyntax (AbstractSyntax const & *as*)

25.199.4.13 void gdcm::network::PresentationContextRQ::SetPresentationContextID (uint8_t *id*)

25.199.4.14 size_t gdcm::network::PresentationContextRQ::Size ( ) const

25.199.4.15 const std::ostream& gdcm::network::PresentationContextRQ::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

- gdcmPresentationContextRQ.h

25.200 gdcm::network::PresentationDataValue Class Reference

PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

```
#include <gdcmPresentationDataValue.h>
```

Public Member Functions

- PresentationDataValue ()
- const std::string & GetBlob () const
- bool GetIsCommand () const
- bool GetIsLastFragment () const
- uint8_t GetMessageHeader () const
- uint8_t GetPresentationContextID () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- std::istream & ReadInto (std::istream &is, std::ostream &os)
- void SetBlob (const std::string &partialblob)
- void SetCommand (bool inCommand)
- void SetDataSet (const DataSet &ds)
- void SetLastFragment (bool inLast)
- void SetMessageHeader (uint8_t messageheader)
- void SetPresentationContextID (uint8_t id)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

Static Public Member Functions

- static DataSet ConcatenatePDVBlobs (const std::vector< PresentationDataValue > &inPDVs)

25.200.1 Detailed Description

PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

25.200.2 Constructor & Destructor Documentation

25.200.2.1 gdcm::network::PresentationDataValue::PresentationDataValue ( )

25.200.3 Member Function Documentation

25.200.3.1 static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobs (const std::vector< PresentationDataValue > & *inPDVs*) [static]

Warning

DataSet will be read as Implicit Little Endian TS

25.200.3.2 const std::string& gdcm::network::PresentationDataValue::GetBlob ( ) const

25.200.3.3 bool gdcm::network::PresentationDataValue::GetIsCommand ( ) const

25.200.3.4 bool gdcm::network::PresentationDataValue::GetIsLastFragment ( ) const

25.200.3.5 uint8_t gdcm::network::PresentationDataValue::GetMessageHeader ( ) const [inline]

25.200.3.6 uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID ( ) const [inline]

25.200.3.7 void gdcm::network::PresentationDataValue::Print (std::ostream & *os*) const

25.200.3.8 std::istream& gdcm::network::PresentationDataValue::Read (std::istream & *is*)

25.200.3.9 std::istream& gdcm::network::PresentationDataValue::ReadInto (std::istream & *is*, std::ostream & *os*)

25.200.3.10 void gdcm::network::PresentationDataValue::SetBlob (const std::string & *partialblob*)

25.200.3.11 void gdcm::network::PresentationDataValue::SetCommand (bool *inCommand*)

25.200.3.12 void gdcm::network::PresentationDataValue::SetDataSet (const DataSet & *ds*)

Set DataSet. Write DataSet in implicit.

Warning

size of dataset should be below maxpdusize

25.200.3.13 void gdcm::network::PresentationDataValue::SetLastFragment (bool *inLast*)

25.200.3.14 void gdcm::network::PresentationDataValue::SetMessageHeader (uint8_t *messageheader*) [inline]

25.200.3.15 void gdcm::network::PresentationDataValue::SetPresentationContextID (uint8_t *id*) [inline]

25.200.3.16 size_t gdcm::network::PresentationDataValue::Size ( ) const

25.200.3.17 const std::ostream& gdcm::network::PresentationDataValue::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

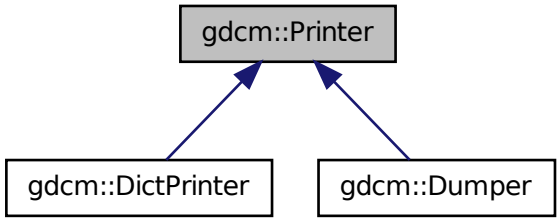
- gdcmPresentationDataValue.h

25.201 gdcm::Printer Class Reference

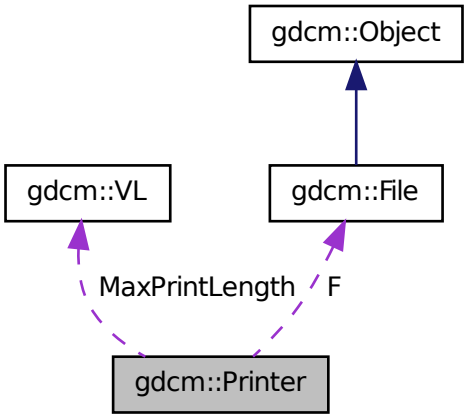
Printer class.

`#include <gdcmPrinter.h>`

Inheritance diagram for gdcm::Printer:



Collaboration diagram for gdcm::Printer:



Public Types

- enum PrintStyles {
  VERBOSE_STYLE = 0,
  CONDENSED_STYLE,
  XML }

Public Member Functions

- Printer ()
- ~Printer ()
- PrintStyles GetPrintStyle () const

  Get PrintStyle value.
- void Print (std::ostream &os)

  Print.
- void PrintDataSet (const DataSet &ds, std::ostream &os, const std::string &s="")

  Print an individual dataset.
- void SetColor (bool c)

  Set color mode or not.
- void SetFile (File const &f)

  Set file.
- void SetStyle (PrintStyles ps)

  Set PrintStyle value.

Protected Member Functions

- VR PrintDataElement (std::ostringstream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, std::ostream &out, std::string const &indent)
- void PrintSQ (const SequenceOfItems ∗sqi, std::ostream &os, std::string const &indent)

Protected Attributes

- const File ∗ F
- VL MaxPrintLength
- PrintStyles PrintStyle

25.201.1 Detailed Description

Printer class.

25.201.2 Member Enumeration Documentation

25.201.2.1 enum gdcm::Printer::PrintStyles

Enumerator:

VERBOSE_STYLE
CONDENSED_STYLE
XML

25.201.3 Constructor & Destructor Documentation

25.201.3.1 gdcm::Printer::Printer ( )

25.201.3.2 gdcm::Printer::∼Printer ( )

25.201.4 Member Function Documentation

25.201.4.1 PrintStyles gdcm::Printer::GetPrintStyle ( ) const [inline]

Get PrintStyle value.

25.201.4.2 void gdcm::Printer::Print (std::ostream & *os*)

Print.

Reimplemented in gdcm::DictPrinter.

25.201.4.3 VR gdcm::Printer::PrintDataElement (std::ostringstream & *os*, const Dicts & *dicts*, const DataSet & *ds*, const DataElement & *de*, std::ostream & *out*, std::string const & *indent*) [protected]

25.201.4.4 void gdcm::Printer::PrintDataSet (const DataSet & *ds*, std::ostream & *os*, const std::string & *s* = " ")

Print an individual dataset.

25.201.4.5 void gdcm::Printer::PrintSQ (const SequenceOfItems * *sqi,* std::ostream & *os,* std::string const & *indent*) [protected]

25.201.4.6 void gdcm::Printer::SetColor (bool *c*)

Set color mode or not.

25.201.4.7 void gdcm::Printer::SetFile (File const & *f*) [inline]

Set file.

25.201.4.8 void gdcm::Printer::SetStyle (PrintStyles *ps*) [inline]

Set PrintStyle value.

25.201.5 Member Data Documentation

25.201.5.1 const File* gdcm::Printer::F [protected]

25.201.5.2 VL gdcm::Printer::MaxPrintLength [protected]

25.201.5.3 PrintStyles gdcm::Printer::PrintStyle [protected]

The documentation for this class was generated from the following file:

- gdcmPrinter.h

25.202 gdcm::PrivateDict Class Reference

Private Dict.

```
#include <gdcmDict.h>
```

Public Member Functions

- PrivateDict ()
- ~PrivateDict ()
- void AddDictEntry (const PrivateTag &tag, const DictEntry &de)
- bool FindDictEntry (const PrivateTag &tag) const
- const DictEntry & GetDictEntry (const PrivateTag &tag) const
- bool IsEmpty () const
- void PrintXML () const
- bool RemoveDictEntry (const PrivateTag &tag)

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &os, const PrivateDict &val)

25.202.1 Detailed Description

Private Dict.

25.202.2 Constructor & Destructor Documentation

25.202.2.1 gdcm::PrivateDict::PrivateDict ( ) [inline]

25.202.2.2 gdcm::PrivateDict::~PrivateDict ( ) [inline]

25.202.3 Member Function Documentation

25.202.3.1 void gdcm::PrivateDict::AddDictEntry (const PrivateTag & *tag,* const DictEntry & *de*) [inline]

References gdcm::DictEntry::GetVM(), gdcm::DictEntry::GetVR(), gdcm::DictEntry::SetVR(), and gdcm::VR::UN.

25.202.3.2 bool gdcm::PrivateDict::FindDictEntry (const PrivateTag & *tag*) const [inline]

25.202.3.3 const DictEntry& gdcm::PrivateDict::GetDictEntry (const PrivateTag & *tag*) const [inline]

25.202.3.4 bool gdcm::PrivateDict::IsEmpty ( ) const [inline]

25.202.3.5 void gdcm::PrivateDict::LoadDefault ( ) [protected]

25.202.3.6 void gdcm::PrivateDict::PrintXML ( ) const [inline]

References gdcm::Tag::GetElement(), gdcm::Tag::GetGroup(), gdcm::DictEntry::GetName(), gdcm::PrivateTag::GetOwner(), gdcm::DictEntry::GetVM(), and gdcm::DictEntry::GetVR().

25.202.3.7 bool gdcm::PrivateDict::RemoveDictEntry (const PrivateTag & *tag*) [inline]

Remove entry 'tag'. Return true on success (element was found and remove). return false if element was not found.

25.202.4 Friends And Related Function Documentation

25.202.4.1 friend class Dicts [friend]

25.202.4.2 std::ostream& operator<< (std::ostream & *os,* const PrivateDict & *val*) [friend]

The documentation for this class was generated from the following file:

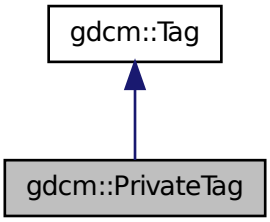
- gdcmDict.h

25.203 gdcm::PrivateTag Class Reference

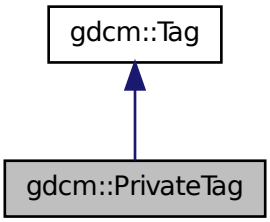
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

`#include <gdcmPrivateTag.h>`

Inheritance diagram for gdcm::PrivateTag:



Collaboration diagram for gdcm::PrivateTag:



Public Member Functions

- PrivateTag (uint16_t group=0, uint16_t element=0, const char ∗owner="")
- const char ∗ GetOwner () const
- bool operator< (const PrivateTag &_val) const
- bool ReadFromCommaSeparatedString (const char ∗str)
- void SetOwner (const char ∗owner)

Friends

- std::ostream & operator<< (std::ostream &_os, const PrivateTag &_val)

25.203.1 Detailed Description

Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

Note

private tag have element value in: [0x10,0xff], for instance 0x0009,0x0000 is NOT a private tag

Examples:

csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, ELSCINT1WaveToText.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, PublicDict.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

25.203.2 Constructor & Destructor Documentation

25.203.2.1 gdcm::PrivateTag::PrivateTag (uint16_t *group* = 0, uint16_t *element* = 0, const char * *owner* = " ") [inline]

25.203.3 Member Function Documentation

25.203.3.1 const char* gdcm::PrivateTag::GetOwner ( ) const [inline]

Examples:

PublicDict.cxx.

Referenced by operator<(), and gdcm::PrivateDict::PrintXML().

25.203.3.2 bool gdcm::PrivateTag::operator< (const **PrivateTag** & *_val*) const [inline]

References GetOwner(), and gdcm::System::StrCaseCmp().

25.203.3.3 bool gdcm::PrivateTag::ReadFromCommaSeparatedString (const char * *str*)

Read from a comma separated string. This is a highly user oriented function, the string should be formated as: 1234,5678 to specify the tag (0x1234,0x5678) The notation comes from the DICOM standard, and is handy to use from a command line program

Reimplemented from gdcm::Tag.

25.203.3.4 void gdcm::PrivateTag::SetOwner (const char * *owner*) [inline]

25.203.4 Friends And Related Function Documentation

25.203.4.1 std::ostream& operator<< (std::ostream & *_os*, const **PrivateTag** & *_val*) [friend]

The documentation for this class was generated from the following file:

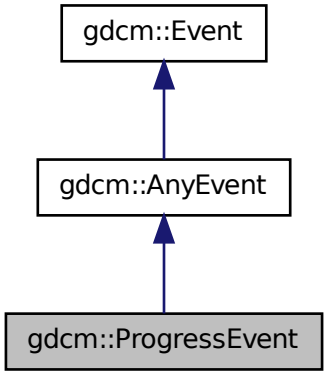
- gdcmPrivateTag.h

25.204 gdcm::ProgressEvent Class Reference

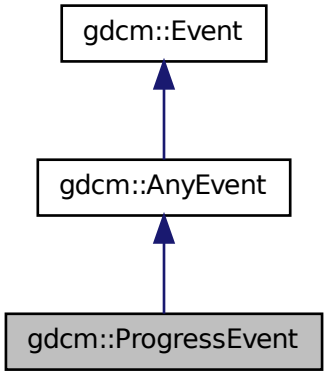
ProgressEvent Special type of event triggered during.

`#include <gdcmProgressEvent.h>`

Inheritance diagram for gdcm::ProgressEvent:



Collaboration diagram for gdcm::ProgressEvent:



Public Types

- typedef ProgressEvent Self

- typedef AnyEvent Superclass

Public Member Functions

- ProgressEvent (double p=0)
- ProgressEvent (const Self &s)
- virtual ∼ProgressEvent ()
- virtual bool CheckEvent (const ::gdcm::Event ∗e) const
- virtual const char ∗ GetEventName () const
- double GetProgress () const
- virtual ::gdcm::Event ∗ MakeObject () const
- void SetProgress (double p)

25.204.1 Detailed Description

ProgressEvent Special type of event triggered during.

See also

AnyEvent

25.204.2 Member Typedef Documentation

25.204.2.1 typedef ProgressEvent gdcm::ProgressEvent::Self

25.204.2.2 typedef AnyEvent gdcm::ProgressEvent::Superclass

25.204.3 Constructor & Destructor Documentation

25.204.3.1 gdcm::ProgressEvent::ProgressEvent (double *p* = 0) [inline]

25.204.3.2 virtual gdcm::ProgressEvent::∼ProgressEvent ( ) [inline], [virtual]

25.204.3.3 gdcm::ProgressEvent::ProgressEvent (const Self & *s*) [inline]

25.204.4 Member Function Documentation

25.204.4.1 virtual bool gdcm::ProgressEvent::CheckEvent (const ::gdcm::Event ∗ *e*) const [inline], [virtual]

25.204.4.2 virtual const char∗ gdcm::ProgressEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

25.204.4.3 double gdcm::ProgressEvent::GetProgress ( ) const [inline]

25.204.4.4 virtual ::gdcm::Event∗ gdcm::ProgressEvent::MakeObject ( ) const [inline], [virtual]

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

25.204.4.5 void gdcm::ProgressEvent::SetProgress (double *p*) [inline]

The documentation for this class was generated from the following file:

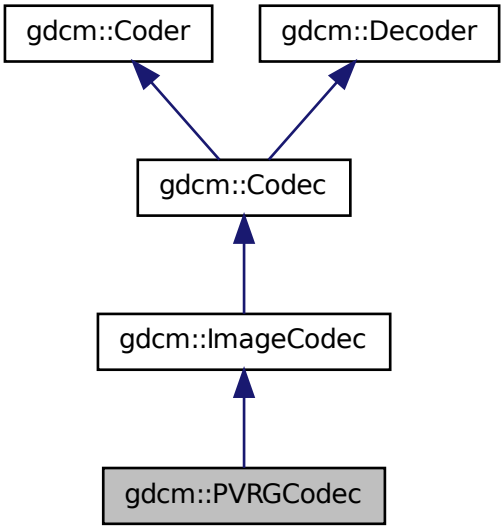
- gdcmProgressEvent.h

25.205 gdcm::PVRGCodec Class Reference

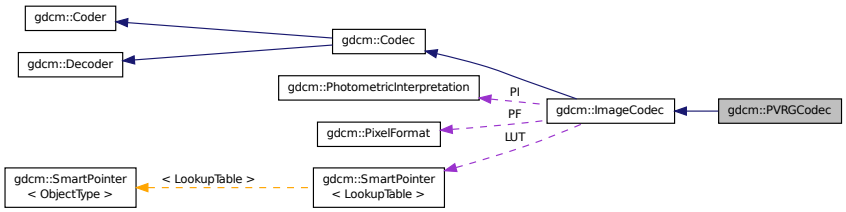
PVRGCodec.

`#include <gdcmPVRGCodec.h>`

Inheritance diagram for gdcm::PVRGCodec:



Collaboration diagram for gdcm::PVRGCodec:



Public Member Functions

- PVRGCodec ()
- ∼PVRGCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Additional Inherited Members

25.205.1 Detailed Description

PVRGCodec.

Note

pvrg is a broken implementation of the JPEG standard. It is known to have a bug in the 16bits lossless implementation of the standard.

In an ideal world, you should not need this codec at all. But to support some broken file such as:

PHILIPS_Gyroscan-12-Jpeg_Extended_Process_2_4.dcm

we have to...

25.205.2 Constructor & Destructor Documentation

25.205.2.1 gdcm::PVRGCodec::PVRGCodec ( )

25.205.2.2 gdcm::PVRGCodec::∼PVRGCodec ( )

25.205.3 Member Function Documentation

25.205.3.1 bool gdcm::PVRGCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.205.3.2 bool gdcm::PVRGCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.205.3.3 bool gdcm::PVRGCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

25.205.3.4 bool gdcm::PVRGCodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

The documentation for this class was generated from the following file:

- gdcmPVRGCodec.h

25.206 gdcm::PythonFilter Class Reference

PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

```
#include <gdcmPythonFilter.h>
```

Public Member Functions

- PythonFilter ()
- ∼PythonFilter ()
- File & GetFile ()
- const File & GetFile () const
- void SetDicts (const Dicts &dicts)
- void SetFile (const File &f)
- PyObject ∗ ToPyObject (const Tag &t) const
- void UseDictAlways (bool use)

25.206.1 Detailed Description

PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

25.206.2 Constructor & Destructor Documentation

25.206.2.1 gdcm::PythonFilter::PythonFilter ( )

25.206.2.2 gdcm::PythonFilter::∼PythonFilter ( )

25.206.3 Member Function Documentation

25.206.3.1 File& gdcm::PythonFilter::GetFile ( ) [inline]

25.206.3.2 const File& gdcm::PythonFilter::GetFile ( ) const [inline]

25.206.3.3 void gdcm::PythonFilter::SetDicts (const Dicts & *dicts*)

25.206.3.4 void gdcm::PythonFilter::SetFile (const File & *f*) [inline]

25.206.3.5 PyObject* gdcm::PythonFilter::ToPyObject (const Tag & *t*) const

25.206.3.6 void gdcm::PythonFilter::UseDictAlways (bool *use*) [inline]

The documentation for this class was generated from the following file:

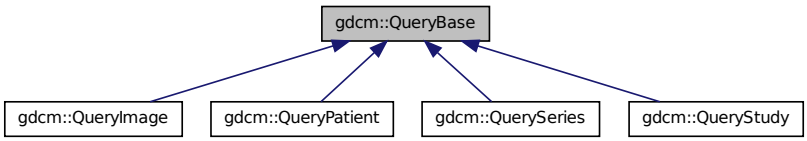
- gdcmPythonFilter.h

25.207 gdcm::QueryBase Class Reference

QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

`#include <gdcmQueryBase.h>`

Inheritance diagram for gdcm::QueryBase:



Public Member Functions

- virtual ~QueryBase ()
- virtual std::vector< Tag > GetAllTags (const ERootType &inRootType) const
- virtual std::string GetName () const =0
- virtual std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const =0
- virtual DataElement GetQueryLevel () const =0
- virtual std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const =0
- virtual std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const =0

25.207.1 Detailed Description

QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

There are four levels of C-FIND and C-MOVE query:

- Patient
- Study

- Series
- Image

Each one has its own required and optional tags. This class provides an interface for getting those tags. This is an interface class.

See 3.4 C 6.1 and 3.4 C 6.2 for the patient and study root query types. These sections define the tags allowed by a particular query. The caller must pass in which root type they want, patient or study. A third root type, Modality Worklist Query, isn't yet supported.

This class (or rather it's derived classes) will be held in the RootQuery types. These query types actually make the dataset, and will use this dataset to list the required, unique, and optional tags for each type of query. This design is somewhat overly complicated, but is kept so that if we ever wanted to try to guess the query type from the given tags, we could do so.

25.207.2 Constructor & Destructor Documentation

25.207.2.1 virtual gdcm::QueryBase::~QueryBase ( ) [inline],[virtual]

25.207.3 Member Function Documentation

25.207.3.1 virtual std::vector<Tag> gdcm::QueryBase::GetAllTags (const ERootType & *inRootType*) const [virtual]

In order to validate a query dataset, just check for the presence of a tag, not it's requirement level in the spec

25.207.3.2 virtual std::string gdcm::QueryBase::GetName ( ) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

25.207.3.3 virtual std::vector<Tag> gdcm::QueryBase::GetOptionalTags (const ERootType & *inRootType*) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

25.207.3.4 virtual DataElement gdcm::QueryBase::GetQueryLevel ( ) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

25.207.3.5 virtual std::vector<Tag> gdcm::QueryBase::GetRequiredTags (const ERootType & *inRootType*) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

25.207.3.6 virtual std::vector<Tag> gdcm::QueryBase::GetUniqueTags (const ERootType & *inRootType*) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

The documentation for this class was generated from the following file:

- gdcmQueryBase.h

25.208 gdcm::QueryFactory Class Reference

QueryFactory.h.

```
#include <gdcmQueryFactory.h>
```

Static Public Member Functions

- static ECharSet GetCharacterFromCurrentLocale ()
- static void ListCharSets (std::ostream &os)

  List all possible CharSet.
- static DataElement ProduceCharacterSetDataElement (const std::vector< ECharSet > &inCharSetType)
- static BaseRootQuery * ProduceQuery (ERootType inRootType, EQueryType inQueryType, EQueryLevel inQueryLevel)

25.208.1 Detailed Description

QueryFactory.h.

Note

contains: a class to produce a query based off of user-entered information

Essentially, this class is used to construct a query based off of user input (typically from the command line; if in code directly, the query itself could just be instantiated)

In theory, could also be used as the interface to validate incoming datasets as belonging to a particular query style

25.208.2 Member Function Documentation

25.208.2.1 static ECharSet gdcm::QueryFactory::GetCharacterFromCurrentLocale ( ) [static]

This function will return the corresponding ECharSet associated with the current locale of the running system (based on the value of locale()).

25.208.2.2 static void gdcm::QueryFactory::ListCharSets (std::ostream & *os*) [static]

List all possible CharSet.

25.208.2.3 static DataElement gdcm::QueryFactory::ProduceCharacterSetDataElement (const std::vector< ECharSet > & *inCharSetType*) [static]

This function will produce the appropriate dataelement given a list of charsets. The first charset will be used directly, while the second and subsequent will be prepended with "ISO2022 ". Redundant character sets are not permitted, so if they are encountered, they will just be skipped. if UTF8 or GB18030 is used, no subsequent character sets will be used if the vector passed in is empty, then the dataelement that's passed out will be empty and Latin1 is the presumed encoding

25.208.2.4 static BaseRootQuery* gdcm::QueryFactory::ProduceQuery (ERootType *inRootType*, EQueryType *inQueryType*, EQueryLevel *inQueryLevel*) [static]

this function will produce a query (basically, a wrapper to a dataset that can validate whether or not the query is a valid cfind/cmove query) and the level of the query (patient, study, series, image). If the user provides an invalid instantiation (ie, study root type, query level of patient), then the result is NULL.

The documentation for this class was generated from the following file:

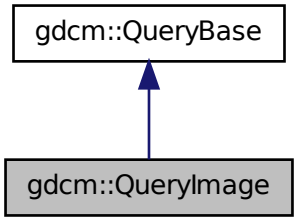
- gdcmQueryFactory.h

25.209 gdcm::QueryImage Class Reference

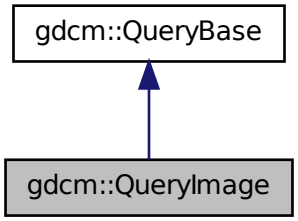
QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

`#include <gdcmQueryImage.h>`

Inheritance diagram for gdcm::QueryImage:



Collaboration diagram for gdcm::QueryImage:



Public Member Functions

- std::string GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

25.209.1 Detailed Description

QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

25.209.2 Member Function Documentation

25.209.2.1 std::string gdcm::QueryImage::GetName ( ) const [inline],[virtual]

Implements gdcm::QueryBase.

25.209.2.2 std::vector<Tag> gdcm::QueryImage::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

25.209.2.3 DataElement gdcm::QueryImage::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

25.209.2.4 std::vector<Tag> gdcm::QueryImage::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

25.209.2.5 std::vector<Tag> gdcm::QueryImage::GetUniqueTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

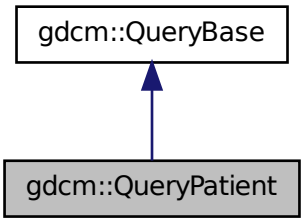
- gdcmQueryImage.h

25.210 gdcm::QueryPatient Class Reference

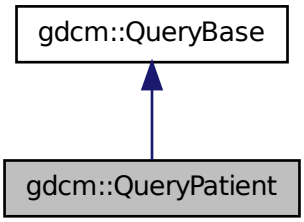
QueryPatient contains: class to construct a patient-based query for c-find and c-move.

```
#include <gdcmQueryPatient.h>
```

Inheritance diagram for gdcm::QueryPatient:



Collaboration diagram for gdcm::QueryPatient:



Public Member Functions

- std::string GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

25.210.1 Detailed Description

QueryPatient contains: class to construct a patient-based query for c-find and c-move.

25.210.2 Member Function Documentation

25.210.2.1 std::string gdcm::QueryPatient::GetName ( ) const [inline], [virtual]

Implements gdcm::QueryBase.

25.210.2.2 std::vector<Tag> gdcm::QueryPatient::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

25.210.2.3 DataElement gdcm::QueryPatient::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

25.210.2.4 std::vector<Tag> gdcm::QueryPatient::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

25.210.2.5 std::vector<Tag> gdcm::QueryPatient::GetUniqueTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

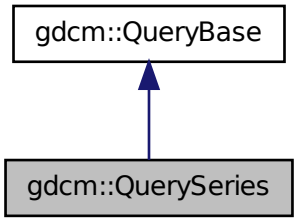
- gdcmQueryPatient.h

25.211 gdcm::QuerySeries Class Reference

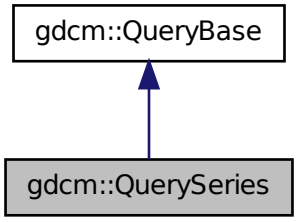
QuerySeries contains: class to construct a series-based query for c-find and c-move.

`#include <gdcmQuerySeries.h>`

Inheritance diagram for gdcm::QuerySeries:



Collaboration diagram for gdcm::QuerySeries:



Public Member Functions

- std::string GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

25.211.1 Detailed Description

QuerySeries contains: class to construct a series-based query for c-find and c-move.

25.211.2 Member Function Documentation

25.211.2.1 std::string gdcm::QuerySeries::GetName ( ) const [inline], [virtual]

Implements gdcm::QueryBase.

25.211.2.2 std::vector<Tag> gdcm::QuerySeries::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

25.211.2.3 DataElement gdcm::QuerySeries::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

25.211.2.4 std::vector<Tag> gdcm::QuerySeries::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

25.211.2.5 std::vector<Tag> gdcm::QuerySeries::GetUniqueTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

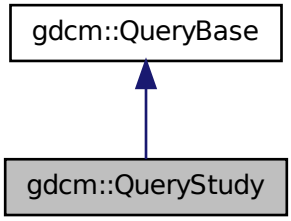
- gdcmQuerySeries.h

25.212 gdcm::QueryStudy Class Reference

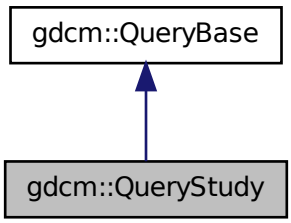
QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

```
#include <gdcmQueryStudy.h>
```

Inheritance diagram for gdcm::QueryStudy:



Collaboration diagram for gdcm::QueryStudy:



Public Member Functions

- std::string GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const

- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

25.212.1 Detailed Description

QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

25.212.2 Member Function Documentation

25.212.2.1 std::string gdcm::QueryStudy::GetName ( ) const [inline], [virtual]

Implements gdcm::QueryBase.

25.212.2.2 std::vector<Tag> gdcm::QueryStudy::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

25.212.2.3 DataElement gdcm::QueryStudy::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

25.212.2.4 std::vector<Tag> gdcm::QueryStudy::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

25.212.2.5 std::vector<Tag> gdcm::QueryStudy::GetUniqueTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

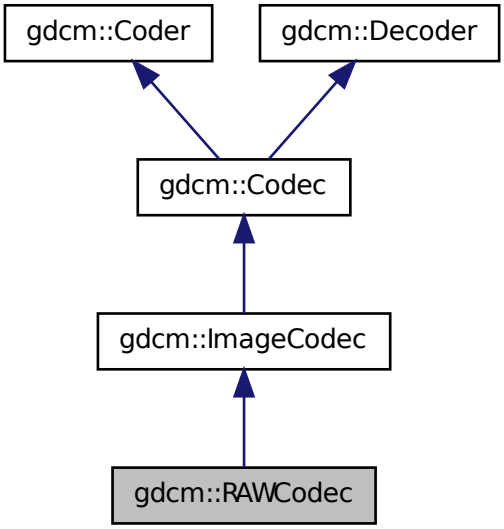
- gdcmQueryStudy.h

25.213 gdcm::RAWCodec Class Reference

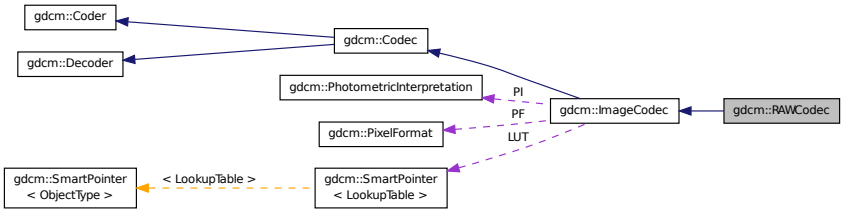
RAWCodec class.

```
#include <gdcmRAWCodec.h>
```

Inheritance diagram for gdcm::RAWCodec:



Collaboration diagram for gdcm::RAWCodec:



Public Member Functions

- RAWCodec ()
- ~RAWCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

Decode.

- bool DecodeBytes (const char ∗inBytes, size_t inBufferLength, char ∗outBytes, size_t inOutBufferLength)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)

Protected Member Functions

- bool Decode (std::istream &is, std::ostream &os)

Additional Inherited Members

25.213.1 Detailed Description

RAWCodec class.

25.213.2 Constructor & Destructor Documentation

25.213.2.1 gdcm::RAWCodec::RAWCodec ( )

25.213.2.2 gdcm::RAWCodec::∼RAWCodec ( )

25.213.3 Member Function Documentation

25.213.3.1 bool gdcm::RAWCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.213.3.2 bool gdcm::RAWCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.213.3.3 bool gdcm::RAWCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

25.213.3.4 bool gdcm::RAWCodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

25.213.3.5 bool gdcm::RAWCodec::Decode (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

25.213.3.6 bool gdcm::RAWCodec::DecodeBytes (const char * *inBytes*, size_t *inBufferLength*, char * *outBytes*, size_t *inOutBufferLength*)

Used by the ImageStreamReader– converts a read in buffer into one with the proper encodings.

25.213.3.7 bool gdcm::RAWCodec::GetHeaderInfo (std::istream & *is*, **TransferSyntax** & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

The documentation for this class was generated from the following file:

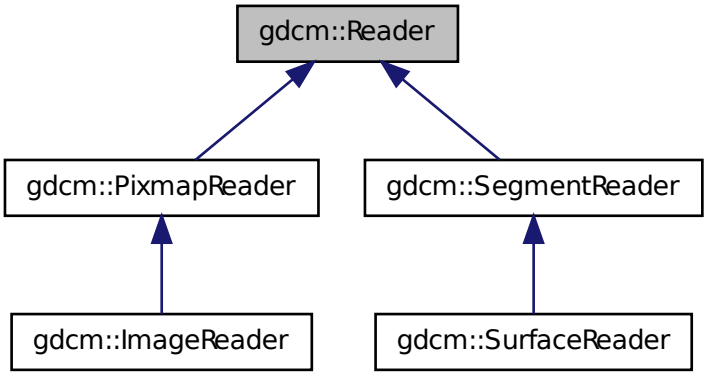
- gdcmRAWCodec.h

25.214 gdcm::Reader Class Reference

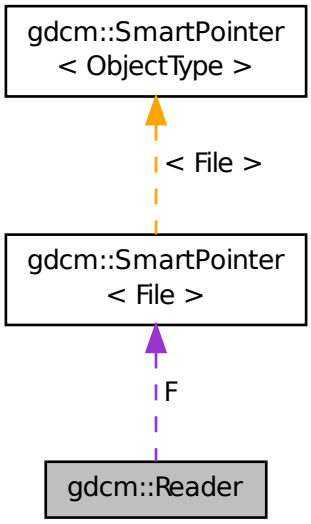
Reader ala DOM (Document Object Model)

`#include <gdcmReader.h>`

Inheritance diagram for gdcm::Reader:



Collaboration diagram for gdcm::Reader:



Public Member Functions

- Reader ()
- virtual ~Reader ()
- bool CanRead () const
- const File & GetFile () const

  Set/Get File.
- File & GetFile ()

  Set/Get File.
- virtual bool Read ()

  Main function to read a file.
- bool ReadSelectedTags (std::set< Tag > const &tags)

  Will only read the specified selected tags.
- bool ReadUpToTag (const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >())
- void SetFile (File &file)

  Set/Get File.
- void SetFileName (const char *filename_native)
- void SetStream (std::istream &input_stream)

  Set the open-ed stream directly.

Protected Member Functions

- std::istream * GetStreamPtr () const

- bool ReadDataSet ()
- bool ReadMetaInformation ()
- bool ReadPreamble ()

Protected Attributes

- SmartPointer< File > F

Friends

- class StreamImageReader

25.214.1 Detailed Description

Reader ala DOM (Document Object Model)

This class is a non-validating reader, it will only performs well- formedness check only, and to some extent catch known error (non well-formed document).

Detailled description here

A DataSet DOES NOT contains group 0x0002 (see FileMetaInformation)

This is really a DataSet reader. This will not make sure the dataset conform to any IOD at all. This is a completely different step. The reasoning was that user could control the IOD there lib would handle and thus we would not be able to read a DataSet if the IOD was not found Instead we separate the reading from the validation.

Note

From GDCM1.x. Users will realize that one feature is missing from this DOM implementation. In GDCM 1.x user used to be able to control the size of the Value to be read. By default it was 0xfff. The main author of GDCM2 thought this was too dangerous and harmful and therefore this feature did not make it into GDCM2

Warning

GDCM will not produce warning for unorder (non-alphabetical order).

See also

Writer FileMetaInformation DataSet File

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadUTF8QtDir.cxx, rle2img.cxx, and TestReader.cxx.

25.214.2 Constructor & Destructor Documentation

25.214.2.1 gdcm::Reader::Reader ( ) [inline]

25.214.2.2 virtual gdcm::Reader::~Reader ( ) [virtual]

25.214.3 Member Function Documentation

25.214.3.1 bool gdcm::Reader::CanRead ( ) const

Test whether this is a DICOM file

Warning

need to call either SetFileName or SetStream first

Examples:

ReadUTF8QtDir.cxx.

25.214.3.2 const File& gdcm::Reader::GetFile ( ) const [inline]

Set/Get File.

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and TestReader.cxx.

25.214.3.3 File& gdcm::Reader::GetFile ( ) [inline]

Set/Get File.

25.214.3.4 std::istream* gdcm::Reader::GetStreamPtr ( ) const [inline],[protected]

25.214.3.5 virtual bool gdcm::Reader::Read ( ) [virtual]

Main function to read a file.

Reimplemented in gdcm::PixmapReader, gdcm::ImageReader, gdcm::SegmentReader, and gdcm::SurfaceReader.

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and TestReader.cxx.

25.214.3.6 bool gdcm::Reader::ReadDataSet ( ) [protected]

25.214.3.7 bool gdcm::Reader::ReadMetaInformation ( ) [protected]

25.214.3.8 bool gdcm::Reader::ReadPreamble ( ) [protected]

25.214.3.9 bool gdcm::Reader::ReadSelectedTags (std::set< Tag > const & *tags*)

Will only read the specified selected tags.

25.214.3.10 bool gdcm::Reader::ReadUpToTag (const Tag & *tag*, std::set< Tag > const & *skiptags* = std::set< Tag >())

Will read only up to Tag

Parameters

tag	and skipping any tag specified in
skiptags	

25.214.3.11 void gdcm::Reader::SetFile (File & *file*) [inline]

Set/Get File.

25.214.3.12 void gdcm::Reader::SetFileName (const char * *filename_native*)

Set the filename to open. This will create a std::ifstream internally See SetStream if you are dealing with different std::istream object

Examples:

ChangeSequenceUltrasound.cxx, CheckBigEndianBug.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, ConvertToQImage.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadMultiTimesException.cxx, ReadUTF8QtDir.cxx, rle2img.cxx, TestReader.cxx, and threadgdcm.cxx.

25.214.3.13 void gdcm::Reader::SetStream (std::istream & *input_stream*) [inline]

Set the open-ed stream directly.

Examples:

ReadUTF8QtDir.cxx.

25.214.4 Friends And Related Function Documentation

25.214.4.1 friend class StreamImageReader [friend]

25.214.5 Member Data Documentation

25.214.5.1 SmartPointer<File> gdcm::Reader::F [protected]

The documentation for this class was generated from the following file:

- gdcmReader.h

25.215 gdcm::Rescaler Class Reference

Rescale class This class is meant to apply the linear transform of Stored Pixel Value to Real World Value. This is mostly found in CT or PET dataset, where the value are stored using one type, but need to be converted to another scale using a linear transform. There are basically two cases: In CT: the linear transform is generally integer based. E.g. the Stored Pixel Type is unsigned short 12bits, but to get Hounsfield unit, one need to apply the linear transform:

$$RWV = 1. * SV - 1024$$

So the best scalar to store the Real World Value will be 16 bits signed type.

```
#include <gdcmRescaler.h>
```

Public Member Functions

- Rescaler ()
- ~Rescaler ()
- PixelFormat::ScalarType ComputeInterceptSlopePixelType ()
- PixelFormat ComputePixelTypeFromMinMax ()
- double GetIntercept () const
- double GetSlope () const
- bool InverseRescale (char *out, const char *in, size_t n)

  Inverse transform.
- bool Rescale (char *out, const char *in, size_t n)

  Direct transform.
- void SetIntercept (double i)

  Set Intercept: used for both direct&inverse transformation.
- void SetMinMaxForPixelType (double min, double max)
- void SetPixelFormat (PixelFormat const &pf)

  Set Pixel Format of input data.
- void SetSlope (double s)

  Set Slope: user for both direct&inverse transformation.
- void SetTargetPixelType (PixelFormat const &targetst)
- void SetUseTargetPixelType (bool b)

  Override default behavior of Rescale.

Protected Member Functions

- template<typename TIn >
  void InverseRescaleFunctionIntoBestFit (char *out, const TIn *in, size_t n)
- template<typename TIn >
  void RescaleFunctionIntoBestFit (char *out, const TIn *in, size_t n)

25.215.1 Detailed Description

Rescale class This class is meant to apply the linear transform of Stored Pixel Value to Real World Value. This is mostly found in CT or PET dataset, where the value are stored using one type, but need to be converted to another scale using a linear transform. There are basically two cases: In CT: the linear transform is generally integer based. E.g. the Stored Pixel Type is unsigned short 12bits, but to get Hounsfield unit, one need to apply the linear transform:

$$RWV = 1. * SV - 1024$$

So the best scalar to store the Real World Value will be 16 bits signed type.

In PET: the linear transform is generally floating point based. Since the dynamic range can be quite high, the Rescale Slope / Rescale Intercept can be changing throughout the Series. So it is important to read all linear transform and deduce the best Pixel Type only at the end (when all the images to be read have been parsed).

Warning

Internally any time a floating point value is found either in the Rescale Slope or the Rescale Intercept it is assumed that the best matching output pixel type is FLOAT64 (in previous implementation it was FLOAT32). Because V-R:DS is closer to a 64bits floating point type FLOAT64 is thus a best matching pixel type for the floating point transformation.

Example: Let say input is FLOAT64, and we want UINT16 as ouput, we would do:

```
  Rescaler ir;
  ir.SetIntercept( 0 );
  ir.SetSlope( 5.6789 );
  ir.SetPixelFormat( FLOAT64 );
  ir.SetMinMaxForPixelType( ((PixelFormat)UINT16).GetMin(), ((PixelFormat)
      UINT16).GetMax() );
  ir.InverseRescale(output,input,numberofbytes );
 *
```

Note

handle floating point transformation back and forth to integer properly (no loss)

See also

Unpacker12Bits

25.215.2 Constructor & Destructor Documentation

25.215.2.1 gdcm::Rescaler::Rescaler ( ) [inline]

25.215.2.2 gdcm::Rescaler::~Rescaler ( ) [inline]

25.215.3 Member Function Documentation

25.215.3.1 PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ( )

Compute the Pixel Format of the output data Used for direct transformation

25.215.3.2 PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ( )

Compute the Pixel Format of the output data Used for inverse transformation

25.215.3.3 double gdcm::Rescaler::GetIntercept ( ) const [inline]

25.215.3.4 double gdcm::Rescaler::GetSlope ( ) const [inline]

25.215.3.5 bool gdcm::Rescaler::InverseRescale (char * *out*, const char * *in*, size_t *n*)

Inverse transform.

25.215.3.6 template<typename TIn > void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (char * *out*, const TIn * *in*, size_t *n*) [protected]

25.215.3.7 bool gdcm::Rescaler::Rescale (char * *out*, const char * *in*, size_t *n*)

Direct transform.

25.215.3.8 template<typename TIn > void gdcm::Rescaler::RescaleFunctionIntoBestFit (char * *out*, const TIn * *in*, size_t *n*) [protected]

25.215.3.9 void gdcm::Rescaler::SetIntercept (double *i*) [inline]

Set Intercept: used for both direct&inverse transformation.

25.215.3.10 void gdcm::Rescaler::SetMinMaxForPixelType (double *min*, double *max*) [inline]

Set target interval for output data. A best match will be computed (if possible) Used for inverse transformation

25.215.3.11 void gdcm::Rescaler::SetPixelFormat (PixelFormat const & *pf*) [inline]

Set Pixel Format of input data.

25.215.3.12 void gdcm::Rescaler::SetSlope (double *s*) [inline]

Set Slope: user for both direct&inverse transformation.

25.215.3.13 void gdcm::Rescaler::SetTargetPixelType (PixelFormat const & *targetst*)

By default (when UseTargetPixelType is false), a best matching Target Pixel Type is computed. However user can override this auto selection by switching UseTargetPixelType:true and also specifying the specifix Target Pixel Type

25.215.3.14 void gdcm::Rescaler::SetUseTargetPixelType (bool *b*)

Override default behavior of Rescale.

The documentation for this class was generated from the following file:

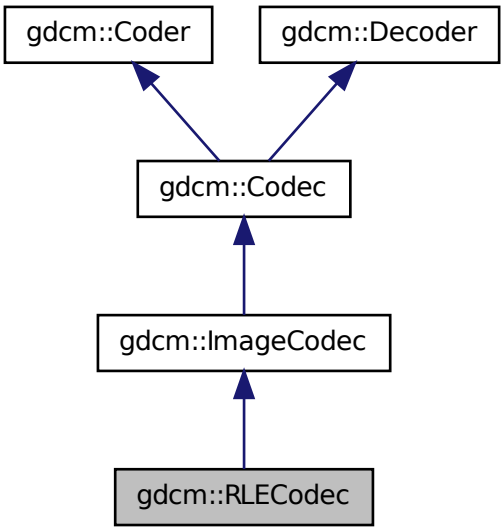
- gdcmRescaler.h

25.216 gdcm::RLECodec Class Reference

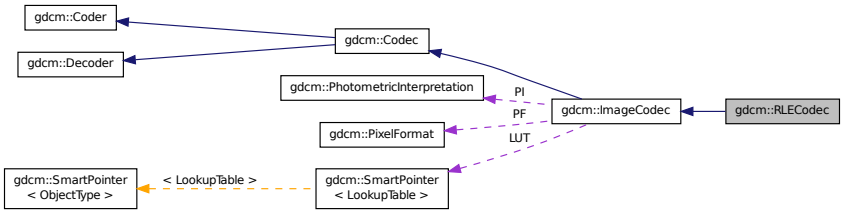
Class to do RLE.

`#include <gdcmRLECodec.h>`

Inheritance diagram for gdcm::RLECodec:



Collaboration diagram for gdcm::RLECodec:



Public Member Functions

- RLECodec ()
- ∼RLECodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- void SetBufferLength (unsigned long l)
- void SetLength (unsigned long l)

Protected Member Functions

- bool Decode (std::istream &is, std::ostream &os)

Additional Inherited Members

25.216.1 Detailed Description

Class to do RLE.

Note

ANSI X3.9 A.4.2 RLE Compression Annex G defines a RLE Compression Transfer Syntax. This transfer Syntax is identified by the UID value "1.2.840.10008.1.2.5". If the object allows multi-frame images in the pixel data field, then each frame shall be encoded separately. Each frame shall be encoded in one and only one Fragment (see PS 3.5.8.2).

25.216.2 Constructor & Destructor Documentation

25.216.2.1 gdcm::RLECodec::RLECodec ( )

25.216.2.2 gdcm::RLECodec::∼RLECodec ( )

25.216.3 Member Function Documentation

25.216.3.1 bool gdcm::RLECodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.216.3.2 bool gdcm::RLECodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.216.3.3 bool gdcm::RLECodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

25.216.3.4 bool gdcm::RLECodec::Decode (DataElement const & *is_*, DataElement & *os*) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

25.216.3.5 bool gdcm::RLECodec::Decode (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

25.216.3.6 unsigned long gdcm::RLECodec::GetBufferLength ( ) const [inline]

25.216.3.7 bool gdcm::RLECodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

25.216.3.8 void gdcm::RLECodec::SetBufferLength (unsigned long *l*) [inline]

25.216.3.9 void gdcm::RLECodec::SetLength (unsigned long *l*) [inline]

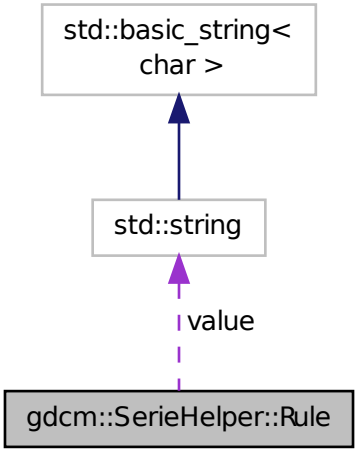
The documentation for this class was generated from the following file:

- gdcmRLECodec.h

25.217 gdcm::SerieHelper::Rule Struct Reference

```
#include <gdcmSerieHelper.h>
```

Collaboration diagram for gdcm::SerieHelper::Rule:



Public Attributes

- uint16_t elem
- uint16_t group
- int op
- std::string value

25.217.1 Member Data Documentation

25.217.1.1 uint16_t gdcm::SerieHelper::Rule::elem

25.217.1.2 uint16_t gdcm::SerieHelper::Rule::group

25.217.1.3 int gdcm::SerieHelper::Rule::op

25.217.1.4 std::string gdcm::SerieHelper::Rule::value

The documentation for this struct was generated from the following file:

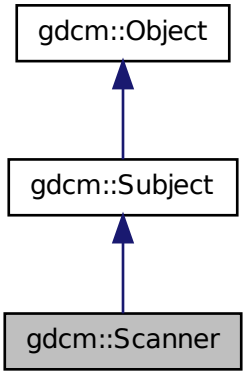
- gdcmSerieHelper.h

25.218 gdcm::Scanner Class Reference

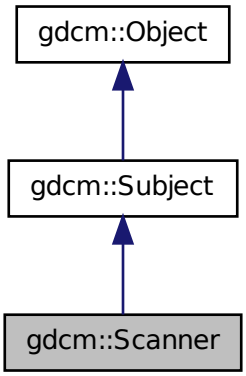
Scanner This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

```
#include <gdcmScanner.h>
```

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for gdcm::Scanner:



Classes

- struct ltstr

Public Types

- typedef MappingType::const_iterator ConstIterator

- typedef std::map< const char *, TagToValue, ltstr > MappingType
- typedef std::map< Tag, const char * > TagToValue
- typedef TagToValue::value_type TagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- Scanner ()
- ~Scanner ()
- void AddPrivateTag (PrivateTag const &t)
- void AddSkipTag (Tag const &t)

  Add a tag that will need to be skipped. Those are root level skip tags.
- void AddTag (Tag const &t)

  Add a tag that will need to be read. Those are root level skip tags.
- ConstIterator Begin () const
- void ClearSkipTags ()
- void ClearTags ()
- ConstIterator End () const
- Directory::FilenamesType GetAllFilenamesFromTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromTagToValue (Tag const &t, const char *valueref) const
- Directory::FilenamesType const & GetFilenames () const
- Directory::FilenamesType GetKeys () const
- TagToValue const & GetMapping (const char *filename) const

  Get the std::map mapping filenames to value for file 'filename'.
- TagToValue const & GetMappingFromTagToValue (Tag const &t, const char *value) const

  See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.
- MappingType const & GetMappings () const

  Mappings are the mapping from a particular tag to the map, mapping filename to value:
- Directory::FilenamesType GetOrderedValues (Tag const &t) const
- const char * GetValue (const char *filename, Tag const &t) const
- ValuesType const & GetValues () const

  Get all the values found (in lexicographic order)
- ValuesType GetValues (Tag const &t) const

  Get all the values found (in lexicographic order) associated with Tag 't'.
- bool IsKey (const char *filename) const
- void Print (std::ostream &os) const

  Print result.
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Static Public Member Functions

- static SmartPointer< Scanner > New ()

  for wrapped language: instanciate a reference counted object

Protected Member Functions

- void ProcessPublicTag (StringFilter &sf, const char *filename)

Friends

- std::ostream & operator<< (std::ostream &_os, const Scanner &s)

25.218.1 Detailed Description

Scanner This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of gdcm::StringFilter

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- ProgressEvent
- StartEvent
- EndEvent

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SimpleScanner.cxx, SortImage.cxx, and VolumeSorter.cxx.

25.218.2 Member Typedef Documentation

25.218.2.1 typedef MappingType::const_iterator gdcm::Scanner::ConstIterator

25.218.2.2 typedef std::map<const char *,TagToValue, ltstr> gdcm::Scanner::MappingType

25.218.2.3 typedef std::map<Tag, const char*> gdcm::Scanner::TagToValue

struct to map a filename to a value Implementation note: all std::map in this class will be using const char * and not std::string since we are pointing to existing std::string (hold in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

25.218.2.4 typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType

25.218.2.5 typedef std::set< std::string > gdcm::Scanner::ValuesType

25.218.3 Constructor & Destructor Documentation

25.218.3.1 gdcm::Scanner::Scanner ( ) [inline]

25.218.3.2 gdcm::Scanner::~Scanner ( )

25.218.4 Member Function Documentation

25.218.4.1 void gdcm::Scanner::AddPrivateTag (PrivateTag const & *t*)

25.218.4.2 void gdcm::Scanner::AddSkipTag (Tag const & *t*)

Add a tag that will need to be skipped. Those are root level skip tags.

25.218.4.3 void gdcm::Scanner::AddTag (Tag const & *t*)

Add a tag that will need to be read. Those are root level skip tags.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SimpleScanner.cxx, SortImage.cxx, and VolumeSorter.cxx.

25.218.4.4 ConstIterator gdcm::Scanner::Begin ( ) const [inline]

25.218.4.5 void gdcm::Scanner::ClearSkipTags ( )

25.218.4.6 void gdcm::Scanner::ClearTags ( )

25.218.4.7 ConstIterator gdcm::Scanner::End ( ) const [inline]

25.218.4.8 Directory::FilenamesType gdcm::Scanner::GetAllFilenamesFromTagToValue (Tag const & *t*, const char * *valueref*) const

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

25.218.4.9 const char* gdcm::Scanner::GetFilenameFromTagToValue (Tag const & *t*, const char * *valueref*) const

Will loop over all files and return the first file where value match the reference value 'valueref'

25.218.4.10 Directory::FilenamesType const& gdcm::Scanner::GetFilenames ( ) const [inline]

25.218.4.11 Directory::FilenamesType gdcm::Scanner::GetKeys ( ) const

Return the list of filename that are key in the internal map, which means those filename were properly parsed

Examples:

VolumeSorter.cxx.

25.218.4.12 TagToValue const& gdcm::Scanner::GetMapping (const char * *filename*) const

Get the std::map mapping filenames to value for file 'filename'.

Examples:

DumpToSQLITE3.cxx, and SimpleScanner.cxx.

25.218.4.13 TagToValue const& gdcm::Scanner::GetMappingFromTagToValue (Tag const & *t*, const char * *value*) const

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

25.218.4.14 MappingType const& gdcm::Scanner::GetMappings ( ) const [inline]

Mappings are the mapping from a particular tag to the map, mapping filename to value:

25.218.4.15 Directory::FilenamesType gdcm::Scanner::GetOrderedValues (Tag const & *t*) const

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

25.218.4.16 const char* gdcm::Scanner::GetValue (const char * *filename*, Tag const & *t*) const

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

25.218.4.17 ValuesType const& gdcm::Scanner::GetValues ( ) const [inline]

Get all the values found (in lexicographic order)

Examples:

SortImage.cxx, and VolumeSorter.cxx.

25.218.4.18 ValuesType gdcm::Scanner::GetValues (Tag const & *t*) const

Get all the values found (in lexicographic order) associated with Tag 't'.

25.218.4.19 bool gdcm::Scanner::IsKey (const char * *filename*) const

Check if filename is a key in the Mapping table. returns true only of file can be found, which means the file was indeed a DICOM file that could be processed

Examples:

DumpToSQLITE3.cxx, and SimpleScanner.cxx.

25.218.4.20 static SmartPointer<Scanner> gdcm::Scanner::New ( ) [inline],[static]

for wrapped language: instanciate a reference counted object

25.218.4.21 void gdcm::Scanner::Print (std::ostream & *os*) const [virtual]

Print result.

Reimplemented from gdcm::Object.

Referenced by gdcm::operator<<().

25.218.4.22 void gdcm::Scanner::ProcessPublicTag (StringFilter & *sf,* const char * *filename*) [protected]

25.218.4.23 bool gdcm::Scanner::Scan (Directory::FilenamesType const & *filenames*)

Start the scan !

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SimpleScanner.cxx, SortImage.cxx, and VolumeSorter.cxx.

25.218.5 Friends And Related Function Documentation

25.218.5.1 std::ostream& operator<< (std::ostream & *_os,* const Scanner & *s*) [friend]

The documentation for this class was generated from the following file:

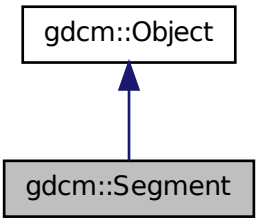
- gdcmScanner.h

25.219 gdcm::Segment Class Reference

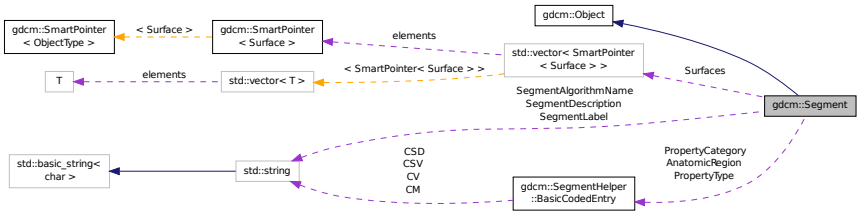
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

```
#include <gdcmSegment.h>
```

Inheritance diagram for gdcm::Segment:



Collaboration diagram for gdcm::Segment:



Public Types

- enum ALGOType {
  MANUAL = 0,
  AUTOMATIC,
  ALGOType_END }
- typedef std::vector
  < SmartPointer< Surface > > SurfaceVector

Public Member Functions

- Segment ()
- virtual ~Segment ()
- void AddSurface (SmartPointer< Surface > surface)
- SegmentHelper::BasicCodedEntry
  const & GetAnatomicRegion () const
- SegmentHelper::BasicCodedEntry & GetAnatomicRegion ()
- SegmentHelper::BasicCodedEntry
  const & GetPropertyCategory () const
- SegmentHelper::BasicCodedEntry & GetPropertyCategory ()
- SegmentHelper::BasicCodedEntry
  const & GetPropertyType () const

- SegmentHelper::BasicCodedEntry & GetPropertyType ()
- const char ∗ GetSegmentAlgorithmName () const
- ALGOType GetSegmentAlgorithmType () const
- const char ∗ GetSegmentDescription () const
- const char ∗ GetSegmentLabel () const
- unsigned short GetSegmentNumber () const
- SmartPointer< Surface > GetSurface (const unsigned int idx=0) const
- unsigned long GetSurfaceCount ()
- SurfaceVector const & GetSurfaces () const
- SurfaceVector & GetSurfaces ()
- void SetAnatomicRegion (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyCategory (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyType (SegmentHelper::BasicCodedEntry const &BSE)
- void SetSegmentAlgorithmName (const char ∗name)
- void SetSegmentAlgorithmType (ALGOType type)
- void SetSegmentAlgorithmType (const char ∗typeStr)
- void SetSegmentDescription (const char ∗description)
- void SetSegmentLabel (const char ∗label)
- void SetSegmentNumber (const unsigned short num)
- void SetSurfaceCount (const unsigned long nb)

Static Public Member Functions

- static ALGOType GetALGOType (const char ∗type)
- static const char ∗ GetALGOTypeString (ALGOType type)

Protected Attributes

- SegmentHelper::BasicCodedEntry AnatomicRegion
- SegmentHelper::BasicCodedEntry PropertyCategory
- SegmentHelper::BasicCodedEntry PropertyType
- std::string SegmentAlgorithmName
- ALGOType SegmentAlgorithmType
- std::string SegmentDescription
- std::string SegmentLabel
- unsigned short SegmentNumber
- unsigned long SurfaceCount
- SurfaceVector Surfaces

Additional Inherited Members

25.219.1 Detailed Description

This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

See also

PS 3.3 C.8.20.2 and C.8.23

25.219.2 Member Typedef Documentation

25.219.2.1 typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector

25.219.3 Member Enumeration Documentation

25.219.3.1 enum gdcm::Segment::ALGOType

Enumerator:

MANUAL

AUTOMATIC

ALGOType_END

25.219.4 Constructor & Destructor Documentation

25.219.4.1 gdcm::Segment::Segment ( )

25.219.4.2 virtual gdcm::Segment::~Segment ( ) [virtual]

25.219.5 Member Function Documentation

25.219.5.1 void gdcm::Segment::AddSurface (SmartPointer< Surface > *surface*)

25.219.5.2 static ALGOType gdcm::Segment::GetALGOType (const char * *type*) [static]

25.219.5.3 static const char* gdcm::Segment::GetALGOTypeString (ALGOType *type*) [static]

25.219.5.4 SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion ( ) const

25.219.5.5 SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ( )

25.219.5.6 SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory ( ) const

25.219.5.7 SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ( )

25.219.5.8 SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType ( ) const

25.219.5.9 SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ( )

25.219.5.10 const char* gdcm::Segment::GetSegmentAlgorithmName ( ) const

25.219.5.11 ALGOType gdcm::Segment::GetSegmentAlgorithmType ( ) const

25.219.5.12 const char* gdcm::Segment::GetSegmentDescription ( ) const

25.219.5.13 const char* gdcm::Segment::GetSegmentLabel ( ) const

25.219.5.14 unsigned short gdcm::Segment::GetSegmentNumber ( ) const

25.219.5.15 SmartPointer< Surface > gdcm::Segment::GetSurface (const unsigned int *idx* = 0) const

25.219.5.16 unsigned long gdcm::Segment::GetSurfaceCount ( )

25.219.5.17 **SurfaceVector** const& gdcm::Segment::GetSurfaces ( ) const

25.219.5.18 **SurfaceVector**& gdcm::Segment::GetSurfaces ( )

25.219.5.19 void gdcm::Segment::SetAnatomicRegion (**SegmentHelper::BasicCodedEntry** const & *BSE*)

25.219.5.20 void gdcm::Segment::SetPropertyCategory (**SegmentHelper::BasicCodedEntry** const & *BSE*)

25.219.5.21 void gdcm::Segment::SetPropertyType (**SegmentHelper::BasicCodedEntry** const & *BSE*)

25.219.5.22 void gdcm::Segment::SetSegmentAlgorithmName (const char * *name*)

25.219.5.23 void gdcm::Segment::SetSegmentAlgorithmType (**ALGOType** *type*)

25.219.5.24 void gdcm::Segment::SetSegmentAlgorithmType (const char * *typeStr*)

25.219.5.25 void gdcm::Segment::SetSegmentDescription (const char * *description*)

25.219.5.26 void gdcm::Segment::SetSegmentLabel (const char * *label*)

25.219.5.27 void gdcm::Segment::SetSegmentNumber (const unsigned short *num*)

25.219.5.28 void gdcm::Segment::SetSurfaceCount (const unsigned long *nb*)

25.219.6 Member Data Documentation

25.219.6.1 **SegmentHelper::BasicCodedEntry** gdcm::Segment::AnatomicRegion [protected]

25.219.6.2 **SegmentHelper::BasicCodedEntry** gdcm::Segment::PropertyCategory [protected]

25.219.6.3 **SegmentHelper::BasicCodedEntry** gdcm::Segment::PropertyType [protected]

25.219.6.4 std::string gdcm::Segment::SegmentAlgorithmName [protected]

25.219.6.5 **ALGOType** gdcm::Segment::SegmentAlgorithmType [protected]

25.219.6.6 std::string gdcm::Segment::SegmentDescription [protected]

25.219.6.7 std::string gdcm::Segment::SegmentLabel [protected]

25.219.6.8 unsigned short gdcm::Segment::SegmentNumber [protected]

25.219.6.9 unsigned long gdcm::Segment::SurfaceCount [protected]

25.219.6.10 **SurfaceVector** gdcm::Segment::Surfaces [protected]

The documentation for this class was generated from the following file:

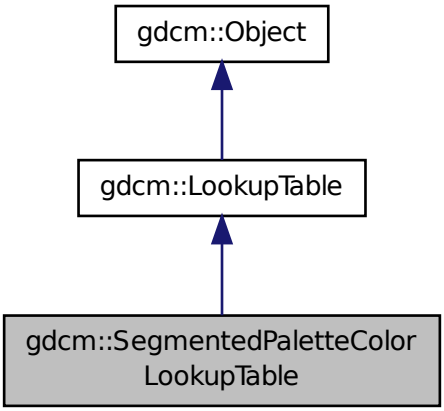
- gdcmSegment.h

25.220 gdcm::SegmentedPaletteColorLookupTable Class Reference

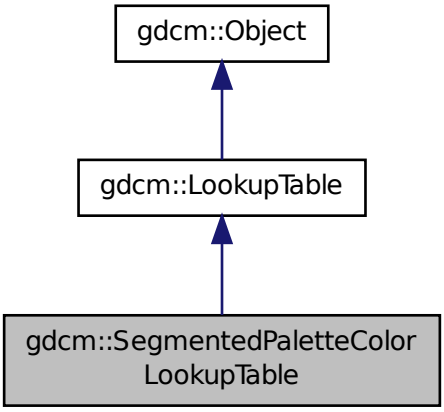
SegmentedPaletteColorLookupTable class.

```
#include <gdcmSegmentedPaletteColorLookupTable.h>
```

Inheritance diagram for gdcm::SegmentedPaletteColorLookupTable:



Collaboration diagram for gdcm::SegmentedPaletteColorLookupTable:



Public Member Functions

- SegmentedPaletteColorLookupTable ()
- ~SegmentedPaletteColorLookupTable ()
- void Print (std::ostream &) const
- void SetLUT (LookupTableType type, const unsigned char *array, unsigned int length)

  Initialize a SegmentedPaletteColorLookupTable.

Additional Inherited Members

25.220.1 Detailed Description

SegmentedPaletteColorLookupTable class.

25.220.2 Constructor & Destructor Documentation

25.220.2.1 gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ( )

25.220.2.2 gdcm::SegmentedPaletteColorLookupTable::~SegmentedPaletteColorLookupTable ( )

25.220.3 Member Function Documentation

25.220.3.1 void gdcm::SegmentedPaletteColorLookupTable::Print (std::ostream &) const [inline],[virtual]

Reimplemented from gdcm::LookupTable.

25.220.3.2 void gdcm::SegmentedPaletteColorLookupTable::SetLUT (LookupTableType *type,* const unsigned char * *array,* unsigned int *length*) [virtual]

Initialize a SegmentedPaletteColorLookupTable.

Reimplemented from gdcm::LookupTable.

The documentation for this class was generated from the following file:

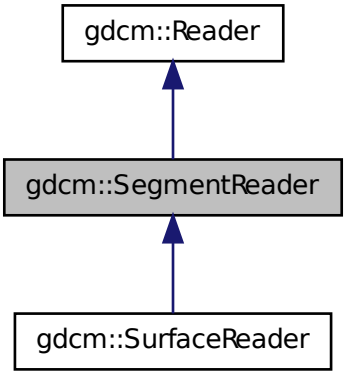
- gdcmSegmentedPaletteColorLookupTable.h

25.221 gdcm::SegmentReader Class Reference

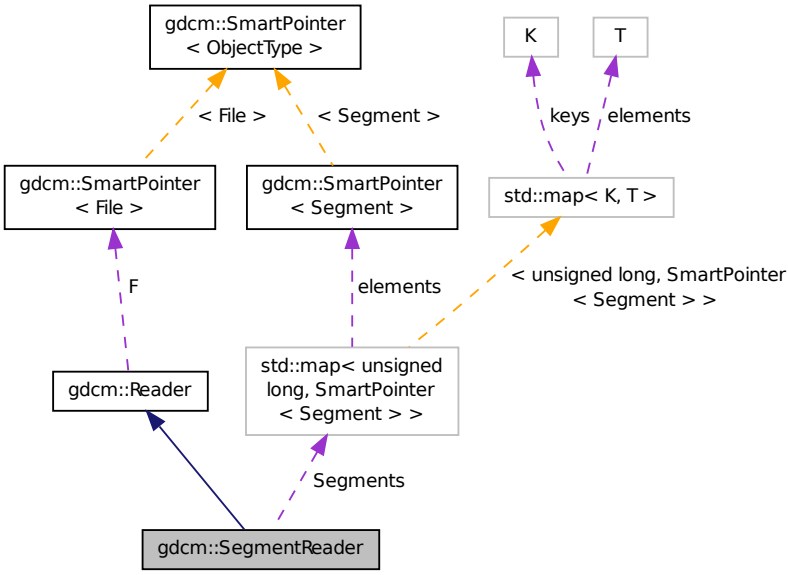
This class defines a segment reader. It reads attributes of group 0x0062.

```
#include <gdcmSegmentReader.h>
```

Inheritance diagram for gdcm::SegmentReader:



Collaboration diagram for gdcm::SegmentReader:



Public Types

- typedef std::vector
  < SmartPointer< Segment > > SegmentVector

Public Member Functions

- SegmentReader ()
- virtual ~SegmentReader ()
- const SegmentVector GetSegments () const
- SegmentVector GetSegments ()
- virtual bool Read ()

  Read.

Protected Types

- typedef std::map< unsigned long, SmartPointer< Segment > > SegmentMap

Protected Member Functions

- bool ReadSegment (const Item &segmentItem, const unsigned int idx)
- bool ReadSegments ()

Protected Attributes

- SegmentMap Segments

25.221.1 Detailed Description

This class defines a segment reader. It reads attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

25.221.2 Member Typedef Documentation

25.221.2.1 typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap [protected]

25.221.2.2 typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector

25.221.3 Constructor & Destructor Documentation

25.221.3.1 gdcm::SegmentReader::SegmentReader ( )

25.221.3.2 virtual gdcm::SegmentReader::~SegmentReader ( ) [virtual]

25.221.4 Member Function Documentation

25.221.4.1 const SegmentVector gdcm::SegmentReader::GetSegments ( ) const

25.221.4.2 SegmentVector gdcm::SegmentReader::GetSegments ( )

25.221.4.3 virtual bool gdcm::SegmentReader::Read ( ) [virtual]

Read.

Reimplemented from gdcm::Reader.

Reimplemented in gdcm::SurfaceReader.

25.221.4.4 bool gdcm::SegmentReader::ReadSegment (const Item & *segmentItem,* const unsigned int *idx*) [protected]

25.221.4.5 bool gdcm::SegmentReader::ReadSegments ( ) [protected]

25.221.5 Member Data Documentation

25.221.5.1 SegmentMap gdcm::SegmentReader::Segments [protected]

The documentation for this class was generated from the following file:

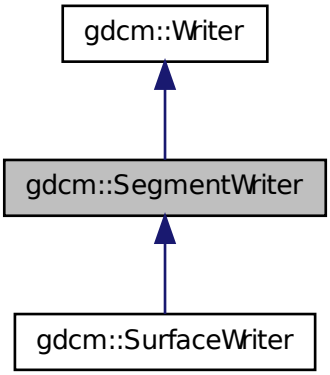
- gdcmSegmentReader.h

25.222 gdcm::SegmentWriter Class Reference

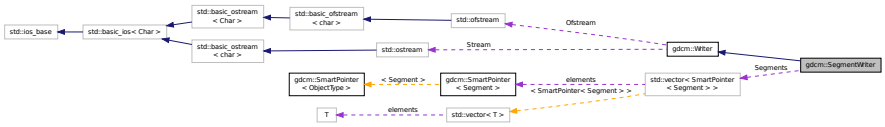
This class defines a segment writer. It writes attributes of group 0x0062.

`#include <gdcmSegmentWriter.h>`

Inheritance diagram for gdcm::SegmentWriter:



Collaboration diagram for gdcm::SegmentWriter:



Public Types

- typedef std::vector
  < SmartPointer< Segment > > SegmentVector

Public Member Functions

- SegmentWriter ()
- virtual ~SegmentWriter ()
- void AddSegment (SmartPointer< Segment > segment)
- unsigned int GetNumberOfSegments () const
- SmartPointer< Segment > GetSegment (const unsigned int idx=0) const
- const SegmentVector & GetSegments () const
- SegmentVector & GetSegments ()
- void SetNumberOfSegments (const unsigned int size)
- void SetSegments (SegmentVector &segments)
- bool Write ()

  Write.

Protected Member Functions

- bool PrepareWrite ()

Protected Attributes

- SegmentVector Segments

25.222.1 Detailed Description

This class defines a segment writer. It writes attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

25.222.2 Member Typedef Documentation

25.222.2.1 typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector

25.222.3 Constructor & Destructor Documentation

25.222.3.1 gdcm::SegmentWriter::SegmentWriter ( )

25.222.3.2 virtual gdcm::SegmentWriter::~SegmentWriter ( ) [virtual]

25.222.4 Member Function Documentation

25.222.4.1 void gdcm::SegmentWriter::AddSegment (SmartPointer< Segment > *segment*)

25.222.4.2 unsigned int gdcm::SegmentWriter::GetNumberOfSegments ( ) const

25.222.4.3 SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (const unsigned int *idx* = 0) const

25.222.4.4 const SegmentVector& gdcm::SegmentWriter::GetSegments ( ) const

25.222.4.5 SegmentVector& gdcm::SegmentWriter::GetSegments ( )

25.222.4.6 bool gdcm::SegmentWriter::PrepareWrite ( ) [protected]

Reimplemented in gdcm::SurfaceWriter.

25.222.4.7 void gdcm::SegmentWriter::SetNumberOfSegments (const unsigned int *size*)

25.222.4.8 void gdcm::SegmentWriter::SetSegments (SegmentVector & *segments*)

25.222.4.9 bool gdcm::SegmentWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::Writer.

Reimplemented in gdcm::SurfaceWriter.

25.222.5 Member Data Documentation

25.222.5.1 SegmentVector gdcm::SegmentWriter::Segments [protected]

The documentation for this class was generated from the following file:

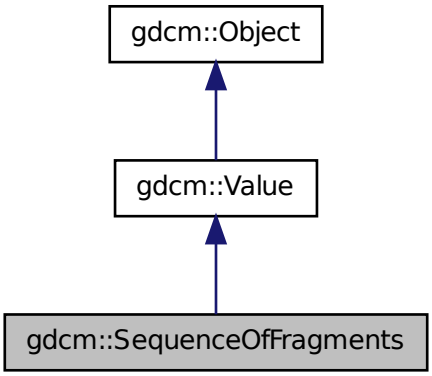
- gdcmSegmentWriter.h

25.223 gdcm::SequenceOfFragments Class Reference

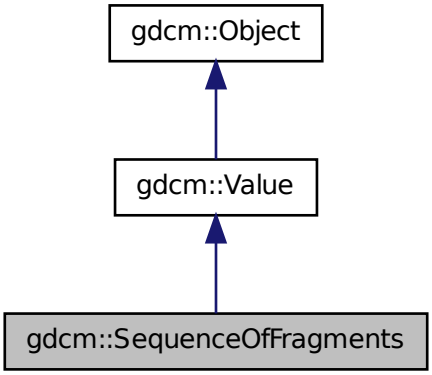
Class to represent a Sequence Of Fragments.

```
#include <gdcmSequenceOfFragments.h>
```

Inheritance diagram for gdcm::SequenceOfFragments:



Collaboration diagram for gdcm::SequenceOfFragments:



Public Types

- typedef FragmentVector::const_iterator ConstIterator
- typedef std::vector< Fragment > FragmentVector
- typedef FragmentVector::iterator Iterator
- typedef FragmentVector::size_type SizeType

Public Member Functions

- SequenceOfFragments ()

  constructor (UndefinedLength by default)
- void AddFragment (Fragment const &item)

  Appends a Fragment to the already added ones.
- Iterator Begin ()
- ConstIterator Begin () const
- void Clear ()

  Clear.
- unsigned long ComputeByteLength () const
- VL ComputeLength () const
- Iterator End ()
- ConstIterator End () const
- bool GetBuffer (char ∗buffer, unsigned long length) const
- bool GetFragBuffer (unsigned int fragNb, char ∗buffer, unsigned long &length) const
- const Fragment & GetFragment (SizeType num) const
- VL GetLength () const

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfFragments () const
- const BasicOffsetTable & GetTable () const
- BasicOffsetTable & GetTable ()
- bool operator== (const Value &val) const
- void Print (std::ostream &os) const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- void SetLength (VL length)

  Sets the actual SQ length.
- template<typename TSwap >
  std::ostream const & Write (std::ostream &os) const
- bool WriteBuffer (std::ostream &os) const

Static Public Member Functions

- static SmartPointer
  < SequenceOfFragments > New ()

25.223.1 Detailed Description

Class to represent a Sequence Of Fragments.

Todo I do not enforce that Sequence of Fragments ends with a SQ end del

Examples:

FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and GetJPEGSamplePrecision.cxx.

25.223.2 Member Typedef Documentation

25.223.2.1 typedef FragmentVector::const_iterator gdcm::SequenceOfFragments::ConstIterator

25.223.2.2 typedef std::vector<Fragment> gdcm::SequenceOfFragments::FragmentVector

25.223.2.3 typedef FragmentVector::iterator gdcm::SequenceOfFragments::Iterator

25.223.2.4 typedef FragmentVector::size_type gdcm::SequenceOfFragments::SizeType

25.223.3 Constructor & Destructor Documentation

25.223.3.1 gdcm::SequenceOfFragments::SequenceOfFragments ( ) [inline]

constructor (UndefinedLength by default)

25.223.4 Member Function Documentation

25.223.4.1 void gdcm::SequenceOfFragments::AddFragment (Fragment const & *item*)

Appends a Fragment to the already added ones.

Examples:

FixBrokenJ2K.cxx.

25.223.4.2 Iterator gdcm::SequenceOfFragments::Begin ( ) [inline]

25.223.4.3 ConstIterator gdcm::SequenceOfFragments::Begin ( ) const [inline]

25.223.4.4 void gdcm::SequenceOfFragments::Clear ( ) [virtual]

Clear.

Implements gdcm::Value.

25.223.4.5 unsigned long gdcm::SequenceOfFragments::ComputeByteLength ( ) const

25.223.4.6 VL gdcm::SequenceOfFragments::ComputeLength ( ) const

25.223.4.7 Iterator gdcm::SequenceOfFragments::End ( ) [inline]

25.223.4.8 ConstIterator gdcm::SequenceOfFragments::End ( ) const [inline]

25.223.4.9 bool gdcm::SequenceOfFragments::GetBuffer (char * *buffer,* unsigned long *length*) const

25.223.4.10 bool gdcm::SequenceOfFragments::GetFragBuffer (unsigned int *fragNb,* char * *buffer,* unsigned long & *length*) const

25.223.4.11 const Fragment& gdcm::SequenceOfFragments::GetFragment (SizeType *num*) const

Examples:

FixBrokenJ2K.cxx, and FixJAIBugJPEGLS.cxx.

25.223.4.12 VL gdcm::SequenceOfFragments::GetLength ( ) const [inline], [virtual]

Returns the SQ length, as read from disk.

Implements gdcm::Value.

25.223.4.13 SizeType gdcm::SequenceOfFragments::GetNumberOfFragments ( ) const

Examples:

FixJAIBugJPEGLS.cxx.

25.223.4.14 const BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) const [inline]

25.223.4.15 BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) [inline]

25.223.4.16 static SmartPointer<SequenceOfFragments> gdcm::SequenceOfFragments::New ( ) [inline], [static]

25.223.4.17 bool gdcm::SequenceOfFragments::operator== (const Value & *val*) const [inline], [virtual]

Implements gdcm::Value.

25.223.4.18 void gdcm::SequenceOfFragments::Print (std::ostream & *os*) const [inline], [virtual]

Reimplemented from gdcm::Object.

25.223.4.19 template<typename TSwap > std::istream& gdcm::SequenceOfFragments::Read (std::istream & *is*) [inline]

References gdcmDebugMacro, gdcmWarningMacro, gdcm::DataElement::GetTag(), gdcm::DataElement::GetVL(), gdcm::Fragment::Read(), gdcm::DataElement::SetByteValue(), and gdcm::Exception::what().

25.223.4.20 void gdcm::SequenceOfFragments::SetLength (VL *length*) [inline], [virtual]

Sets the actual SQ length.

Implements gdcm::Value.

25.223.4.21 template<typename TSwap > std::ostream const& gdcm::SequenceOfFragments::Write (std::ostream & *os*) const [inline]

References gdcm::VL::Write(), and gdcm::Tag::Write().

25.223.4.22 bool gdcm::SequenceOfFragments::WriteBuffer (std::ostream & *os*) const

Examples:

GetJPEGSamplePrecision.cxx.

The documentation for this class was generated from the following file:

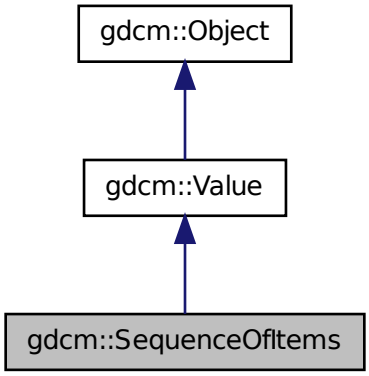
- gdcmSequenceOfFragments.h

25.224 gdcm::SequenceOfItems Class Reference

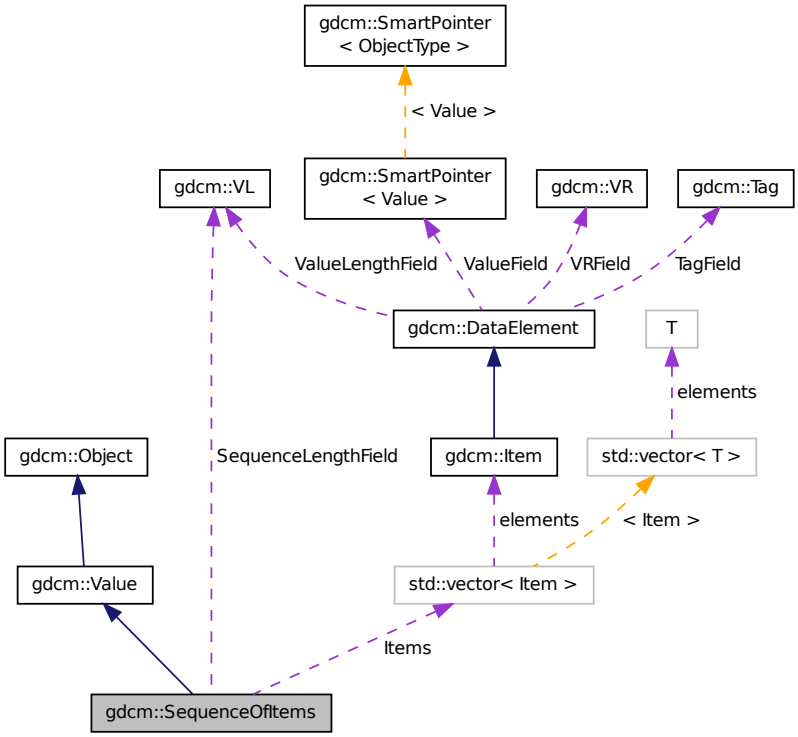
Class to represent a Sequence Of Items (value representation : SQ)

`#include <gdcmSequenceOfItems.h>`

Inheritance diagram for gdcm::SequenceOfItems:



Collaboration diagram for gdcm::SequenceOfItems:



Public Types

- typedef ItemVector::const_iterator ConstIterator
- typedef std::vector< Item > ItemVector
- typedef ItemVector::iterator Iterator
- typedef ItemVector::size_type SizeType

Public Member Functions

- SequenceOfItems ()

  constructor (UndefinedLength by default)
- void AddItem (Item const &item)

  Appends an Item to the already added ones.
- Iterator Begin ()
- ConstIterator Begin () const
- void Clear ()
- template<typename TDE >
  VL ComputeLength () const
- Iterator End ()
- ConstIterator End () const

- bool FindDataElement (const Tag &t) const
- const Item & GetItem (SizeType position) const
- Item & GetItem (SizeType position)
- VL GetLength () const

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfItems () const
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- SequenceOfItems & operator= (const SequenceOfItems &val)
- bool operator== (const Value &val) const
- void Print (std::ostream &os) const
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is)
- void SetLength (VL length)

  Sets the actual SQ length.
- void SetLengthToUndefined ()

  Properly set the Sequence of Item to be undefined length.
- void SetNumberOfItems (SizeType n)
- template<typename TDE , typename TSwap >
  std::ostream const & Write (std::ostream &os) const

Static Public Member Functions

- static SmartPointer
  < SequenceOfItems > New ()

Public Attributes

- ItemVector Items

  Vector of Sequence Items.
- VL SequenceLengthField

  Total length of the Sequence (or 0xffffffff) if undefined.

25.224.1 Detailed Description

Class to represent a Sequence Of Items (value representation : SQ)

- a Value Representation for Data Elements that contains a sequence of Data Sets.
- Sequence of Item allows for Nested Data Sets

See PS 3.5, 7.4.6 Data Element Type Within a Sequence

Note

SEQUENCE OF ITEMS (VALUE REPRESENTATION SQ) A Value Representation for Data Elements that contain a sequence of Data Sets. Sequence of Items allows for Nested Data Sets.

Examples:

DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, and ReadExplicitLengthSQIVR.cxx.

25.224.2 Member Typedef Documentation

25.224.2.1 typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator

25.224.2.2 typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector

25.224.2.3 typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator

25.224.2.4 typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType

25.224.3 Constructor & Destructor Documentation

25.224.3.1 gdcm::SequenceOfItems::SequenceOfItems ( ) [inline]

constructor (UndefinedLength by default)

25.224.4 Member Function Documentation

25.224.4.1 void gdcm::SequenceOfItems::AddItem (Item const & *item*)

Appends an Item to the already added ones.

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

25.224.4.2 Iterator gdcm::SequenceOfItems::Begin ( ) [inline]

25.224.4.3 ConstIterator gdcm::SequenceOfItems::Begin ( ) const [inline]

25.224.4.4 void gdcm::SequenceOfItems::Clear ( ) [inline], [virtual]

Implements gdcm::Value.

25.224.4.5 template<typename TDE > VL gdcm::SequenceOfItems::ComputeLength ( ) const

25.224.4.6 Iterator gdcm::SequenceOfItems::End ( ) [inline]

25.224.4.7 ConstIterator gdcm::SequenceOfItems::End ( ) const [inline]

25.224.4.8 bool gdcm::SequenceOfItems::FindDataElement (const Tag & *t*) const

25.224.4.9 const Item& gdcm::SequenceOfItems::GetItem (SizeType *position*) const

Examples:

ChangeSequenceUltrasound.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

25.224.4.10 Item& gdcm::SequenceOfItems::GetItem (SizeType *position*)

25.224.4.11 VL gdcm::SequenceOfItems::GetLength ( ) const [inline], [virtual]

Returns the SQ length, as read from disk.

Implements gdcm::Value.

25.224.4.12 SizeType gdcm::SequenceOfItems::GetNumberOfItems ( ) const [inline]

Examples:

ChangeSequenceUltrasound.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, and LargeVRDSExplicit.cxx.

25.224.4.13 bool gdcm::SequenceOfItems::IsUndefinedLength ( ) const [inline]

return if Value Length if of undefined length

25.224.4.14 static SmartPointer<SequenceOfItems> gdcm::SequenceOfItems::New ( ) [inline], [static]

25.224.4.15 SequenceOfItems& gdcm::SequenceOfItems::operator= (const SequenceOfItems & *val*) [inline]

References Items, and SequenceLengthField.

25.224.4.16 bool gdcm::SequenceOfItems::operator== (const Value & *val*) const [inline], [virtual]

Implements gdcm::Value.

References Items, and SequenceLengthField.

25.224.4.17 void gdcm::SequenceOfItems::Print (std::ostream & *os*) const [inline], [virtual]

Reimplemented from gdcm::Object.

25.224.4.18 template<typename TDE , typename TSwap > std::istream& gdcm::SequenceOfItems::Read (std::istream & *is*) [inline]

Examples:

ReadExplicitLengthSQIVR.cxx.

References gdcm::Item::Clear(), gdcmDebugMacro, gdcmWarningMacro, gdcm::Exception::GetDescription(), gdcm::Item::GetNestedDataSet(), gdcm::DataElement::GetTag(), gdcm::DataElement::GetVL(), gdcm::Item::Read(), and gdcm::DataSet::Size().

25.224.4.19 void gdcm::SequenceOfItems::SetLength (VL *length*) [inline], [virtual]

Sets the actual SQ length.

Implements gdcm::Value.

Examples:

ReadExplicitLengthSQIVR.cxx.

25.224.4.20 void gdcm::SequenceOfItems::SetLengthToUndefined ( )

Properly set the Sequence of Item to be undefined length.

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

25.224.4.21 void gdcm::SequenceOfItems::SetNumberOfItems (SizeType *n*) [inline]

25.224.4.22 template<typename TDE , typename TSwap > std::ostream const& gdcm::SequenceOfItems::Write (std::ostream & *os*) const [inline]

References gdcm::VL::Write(), and gdcm::Tag::Write().

25.224.5 Member Data Documentation

25.224.5.1 ItemVector gdcm::SequenceOfItems::Items

Vector of Sequence Items.

Referenced by operator=(), and operator==().

25.224.5.2 VL gdcm::SequenceOfItems::SequenceLengthField

Total length of the Sequence (or 0xffffffff) if undefined.

Referenced by operator=(), and operator==().

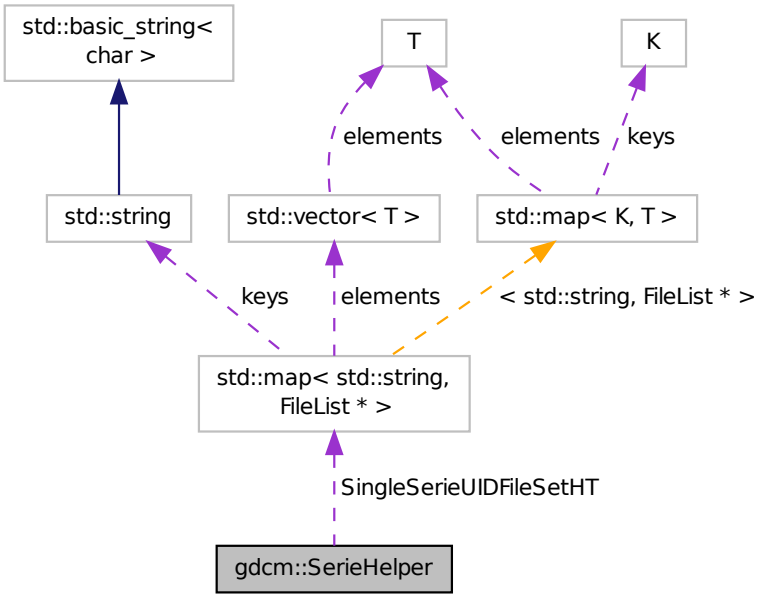
The documentation for this class was generated from the following file:

- gdcmSequenceOfItems.h

25.225 gdcm::SerieHelper Class Reference

```
#include <gdcmSerieHelper.h>
```

Collaboration diagram for gdcm::SerieHelper:



Classes

- struct Rule

Public Member Functions

- SerieHelper ()
- ~SerieHelper ()
- void AddRestriction (const std::string &tag)
- void AddRestriction (uint16_t group, uint16_t elem, std::string const &value, int op)
- void Clear ()
- void CreateDefaultUniqueSeriesIdentifier ()
- std::string CreateUniqueSeriesIdentifier (File *inFile)
- FileList * GetFirstSingleSerieUIDFileSet ()
- FileList * GetNextSingleSerieUIDFileSet ()
- void OrderFileList (FileList *fileSet)
- void SetDirectory (std::string const &dir, bool recursive=false)
- void SetLoadMode (int)
- void SetUseSeriesDetails (bool useSeriesDetails)

Protected Types

- typedef std::vector< Rule > SerieRestrictions
- typedef std::map< std::string, FileList * > SingleSerieUIDFileSetmap

Protected Member Functions

- bool AddFile (FileWithName &header)
- void AddFileName (std::string const &filename)
- void AddRestriction (const Tag &tag)
- bool FileNameOrdering (FileList *fileList)
- bool ImagePositionPatientOrdering (FileList *fileSet)
- bool UserOrdering (FileList *fileSet)

Protected Attributes

- SingleSerieUIDFileSetmap::iterator ItFileSetHt
- SingleSerieUIDFileSetmap SingleSerieUIDFileSetHT

25.225.1 Detailed Description

DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disapear soon, you've been warned.

Instead see gdcm::ImageHelper or gdcm::IPPSorter

25.225.2 Member Typedef Documentation

25.225.2.1 typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions [protected]

25.225.2.2 typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap [protected]

25.225.3 Constructor & Destructor Documentation

25.225.3.1 gdcm::SerieHelper::SerieHelper ( )

25.225.3.2 gdcm::SerieHelper::~SerieHelper ( )

25.225.4 Member Function Documentation

25.225.4.1 bool gdcm::SerieHelper::AddFile (FileWithName & *header*) [protected]

25.225.4.2 void gdcm::SerieHelper::AddFileName (std::string const & *filename*) [protected]

25.225.4.3 void gdcm::SerieHelper::AddRestriction (const std::string & *tag*)

25.225.4.4 void gdcm::SerieHelper::AddRestriction (uint16_t *group,* uint16_t *elem,* std::string const & *value,* int *op*)

25.225.4.5 void gdcm::SerieHelper::AddRestriction (const Tag & *tag*) [protected]

25.225.4.6 void gdcm::SerieHelper::Clear ( )

25.225.4.7 void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ( )

25.225.4.8 std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (File * *inFile*)

25.225.4.9 bool gdcm::SerieHelper::FileNameOrdering (FileList * *fileList*) [protected]

25.225.4.10 FileList* gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ( )

25.225.4.11 FileList* gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ( )

25.225.4.12 bool gdcm::SerieHelper::ImagePositionPatientOrdering (FileList * *fileSet*) [protected]

25.225.4.13 void gdcm::SerieHelper::OrderFileList (FileList * *fileSet*)

25.225.4.14 void gdcm::SerieHelper::SetDirectory (std::string const & *dir,* bool *recursive* = false)

25.225.4.15 void gdcm::SerieHelper::SetLoadMode (int) [inline]

25.225.4.16 void gdcm::SerieHelper::SetUseSeriesDetails (bool *useSeriesDetails*)

25.225.4.17 bool gdcm::SerieHelper::UserOrdering (FileList * *fileSet*) [protected]

25.225.5 Member Data Documentation

25.225.5.1 SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt [protected]

25.225.5.2 SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT [protected]

The documentation for this class was generated from the following file:

- gdcmSerieHelper.h

25.226 gdcm::Series Class Reference

Series.

```
#include <gdcmSeries.h>
```

Public Member Functions

- Series ()

25.226.1 Detailed Description

Series.

25.226.2 Constructor & Destructor Documentation

25.226.2.1 gdcm::Series::Series ( ) [inline]

The documentation for this class was generated from the following file:

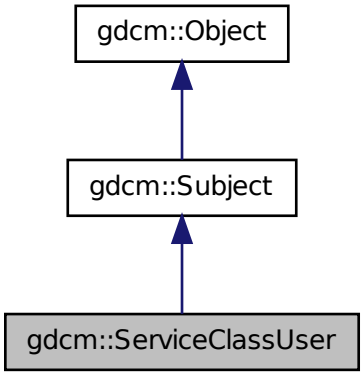
- gdcmSeries.h

25.227 gdcm::ServiceClassUser Class Reference

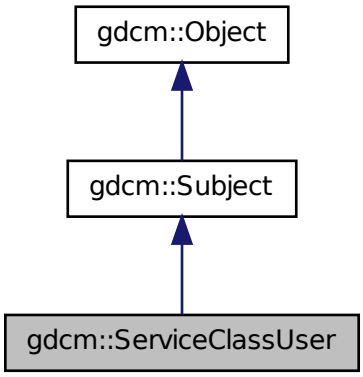
ServiceClassUser.

`#include <gdcmServiceClassUser.h>`

Inheritance diagram for gdcm::ServiceClassUser:



Collaboration diagram for gdcm::ServiceClassUser:



Public Member Functions

- ServiceClassUser ()
- ~ServiceClassUser ()
- const char * GetAETitle () const
- const char * GetCalledAETitle () const
- time_t GetTimeout () const
- bool InitializeConnection ()
- bool SendEcho ()

  C-ECHO.
- bool SendFind (const BaseRootQuery *query, std::vector< DataSet > &retDatasets)

  C-FIND a query, return result are in retDatasets.
- bool SendMove (const BaseRootQuery *query, const char *outputdir)

  Execute a C-MOVE, based on query, return files are written in outputdir.
- bool SendMove (const BaseRootQuery *query, std::vector< DataSet > &retDatasets)

  Execute a C-MOVE, based on query, returned dataset are Implicit.
- bool SendMove (const BaseRootQuery *query, std::vector< File > &retFile)

  Execute a C-MOVE, based on query, returned Files are stored in vector.
- bool SendStore (const char *filename)

  Execute a C-STORE on file on disk, named filename.
- bool SendStore (File const &file)
- bool SendStore (DataSet const &ds)

  Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.
- void SetAETitle (const char *aetitle)

  set calling ae title
- void SetCalledAETitle (const char *aetitle)

  set called ae title
- void SetHostname (const char *hostname)

Set the name of the called hostname (hostname or IP address)

- void SetPort (uint16_t port)

  Set port of remote host (called application)

- void SetPortSCP (uint16_t portscp)

  Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)

- void SetPresentationContexts (std::vector< PresentationContext > const &pcs)

  Set the Presentation Context used for the Association.

- void SetTimeout (time_t t)

  set/get Timeout

- bool StartAssociation ()

  Start the association. Need to call SetPresentationContexts before.

- bool StopAssociation ()

  Stop the running association.

25.227.1 Detailed Description

ServiceClassUser.

Examples:

CStoreQtProgress.cxx.

25.227.2 Constructor & Destructor Documentation

25.227.2.1 gdcm::ServiceClassUser::ServiceClassUser ( )

Construct a SCU with default:

- hostname = localhost
- port = 104

25.227.2.2 gdcm::ServiceClassUser::~ServiceClassUser ( )

25.227.3 Member Function Documentation

25.227.3.1 const char* gdcm::ServiceClassUser::GetAETitle ( ) const

25.227.3.2 const char* gdcm::ServiceClassUser::GetCalledAETitle ( ) const

25.227.3.3 time_t gdcm::ServiceClassUser::GetTimeout ( ) const

25.227.3.4 bool gdcm::ServiceClassUser::InitializeConnection ( )

Will try to connect This will setup the actual timeout used during the whole connection time. Need to call SetTimeout first

Examples:

CStoreQtProgress.cxx.

25.227.3.5 bool gdcm::ServiceClassUser::SendEcho ( )

C-ECHO.

25.227.3.6 bool gdcm::ServiceClassUser::SendFind (const BaseRootQuery * *query,* std::vector< DataSet > & *retDatasets*)

C-FIND a query, return result are in retDatasets.

25.227.3.7 bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * *query,* const char * *outputdir*)

Execute a C-MOVE, based on query, return files are written in outputdir.

25.227.3.8 bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * *query,* std::vector< DataSet > & *retDatasets*)

Execute a C-MOVE, based on query, returned dataset are Implicit.

25.227.3.9 bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * *query,* std::vector< File > & *retFile*)

Execute a C-MOVE, based on query, returned Files are stored in vector.

25.227.3.10 bool gdcm::ServiceClassUser::SendStore (const char * *filename*)

Execute a C-STORE on file on disk, named filename.

Examples:

CStoreQtProgress.cxx.

25.227.3.11 bool gdcm::ServiceClassUser::SendStore (File const & *file*)

Execute a C-STORE on a File, the transfer syntax used for the query is based on the file.

25.227.3.12 bool gdcm::ServiceClassUser::SendStore (DataSet const & *ds*)

Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.

25.227.3.13 void gdcm::ServiceClassUser::SetAETitle (const char * *aetitle*)

set calling ae title

25.227.3.14 void gdcm::ServiceClassUser::SetCalledAETitle (const char * *aetitle*)

set called ae title

Examples:

CStoreQtProgress.cxx.

25.227.3.15 void gdcm::ServiceClassUser::SetHostname (const char * *hostname*)

Set the name of the called hostname (hostname or IP address)

Examples:

CStoreQtProgress.cxx.

25.227.3.16 void gdcm::ServiceClassUser::SetPort (uint16_t *port*)

Set port of remote host (called application)

Examples:

CStoreQtProgress.cxx.

25.227.3.17 void gdcm::ServiceClassUser::SetPortSCP (uint16_t *portscp*)

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)

25.227.3.18 void gdcm::ServiceClassUser::SetPresentationContexts (std::vector< PresentationContext > const & *pcs*)

Set the Presentation Context used for the Association.

Examples:

CStoreQtProgress.cxx.

25.227.3.19 void gdcm::ServiceClassUser::SetTimeout (time_t *t*)

set/get Timeout

Examples:

CStoreQtProgress.cxx.

25.227.3.20 bool gdcm::ServiceClassUser::StartAssociation ( )

Start the association. Need to call SetPresentationContexts before.

Examples:

CStoreQtProgress.cxx.

25.227.3.21 bool gdcm::ServiceClassUser::StopAssociation ( )

Stop the running association.

Examples:

CStoreQtProgress.cxx.

The documentation for this class was generated from the following file:

- gdcmServiceClassUser.h

25.228 gdcm::SHA1 Class Reference

Class for SHA1.

```
#include <gdcmSHA1.h>
```

Public Member Functions

- SHA1 ()
- ∼SHA1 ()

Static Public Member Functions

- static bool Compute (const char ∗buffer, unsigned long buf_len, char digest_str[20 ∗2+1])
- static bool ComputeFile (const char ∗filename, char digest_str[20 ∗2+1])

25.228.1 Detailed Description

Class for SHA1.

Warning

this class is able to pick from one implementation:

1. the one from OpenSSL (when GDCM_USE_SYSTEM_OPENSSL is turned ON)

In all other cases it will return an error

25.228.2 Constructor & Destructor Documentation

25.228.2.1 gdcm::SHA1::SHA1 ( )

25.228.2.2 gdcm::SHA1::∼SHA1 ( )

25.228.3 Member Function Documentation

25.228.3.1 static bool gdcm::SHA1::Compute (const char ∗ *buffer,* unsigned long *buf_len,* char *digest_str[20 ∗2+1]*) [static]

25.228.3.2 static bool gdcm::SHA1::ComputeFile (const char * *filename*, char *digest_str[20 * 2+1]*) [static]

The documentation for this class was generated from the following file:

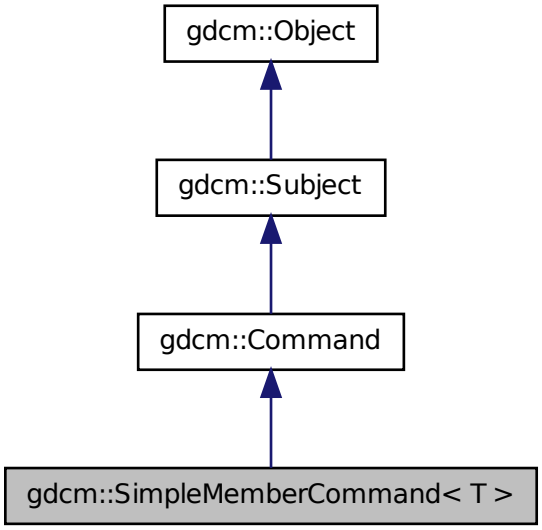
- gdcmSHA1.h

25.229 gdcm::SimpleMemberCommand< T > Class Template Reference

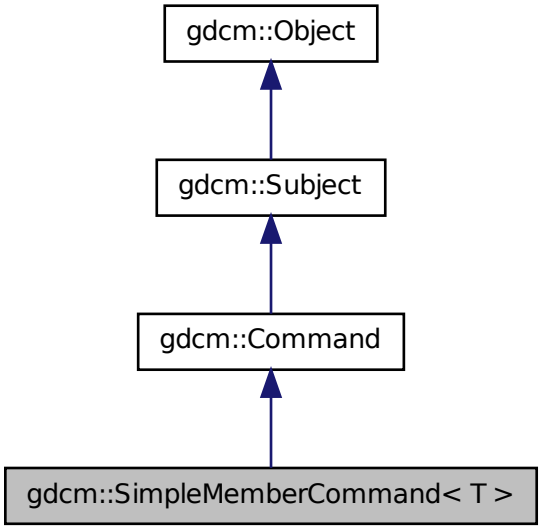
Command subclass that calls a pointer to a member function.

`#include <gdcmCommand.h>`

Inheritance diagram for gdcm::SimpleMemberCommand< T >:



Collaboration diagram for gdcm::SimpleMemberCommand< T >:



Public Types

- typedef SimpleMemberCommand Self
- typedef void(T::∗ TMemberFunctionPointer)()

Public Member Functions

- virtual void Execute (Subject ∗, const Event &)
- virtual void Execute (const Subject ∗, const Event &)
- void SetCallbackFunction (T ∗object, TMemberFunctionPointer memberFunction)

Static Public Member Functions

- static SmartPointer
  < SimpleMemberCommand > New ()

Protected Member Functions

- SimpleMemberCommand ()
- virtual ∼SimpleMemberCommand ()

Protected Attributes

- TMemberFunctionPointer m_MemberFunction
- T ∗ m_This

25.229.1 Detailed Description

template<typename T>class gdcm::SimpleMemberCommand< T >

Command subclass that calls a pointer to a member function.

SimpleMemberCommand calls a pointer to a member function with no arguments.

25.229.2 Member Typedef Documentation

25.229.2.1 template<typename T > typedef SimpleMemberCommand gdcm::SimpleMemberCommand< T >::Self

Standard class typedefs.

25.229.2.2 template<typename T > typedef void(T::∗ gdcm::SimpleMemberCommand< T >::TMemberFunctionPointer)()

A method callback.

25.229.3 Constructor & Destructor Documentation

25.229.3.1 template<typename T > gdcm::SimpleMemberCommand< T >::SimpleMemberCommand ( ) [inline], [protected]

Referenced by gdcm::SimpleMemberCommand< T >::New().

25.229.3.2 template<typename T > virtual gdcm::SimpleMemberCommand< T >::~SimpleMemberCommand ( ) [inline], [protected], [virtual]

25.229.4 Member Function Documentation

25.229.4.1 template<typename T > virtual void gdcm::SimpleMemberCommand< T >::Execute (Subject ∗ , const Event &) [inline], [virtual]

Invoke the callback function.

Implements gdcm::Command.

References gdcm::SimpleMemberCommand< T >::m_MemberFunction.

25.229.4.2 template<typename T > virtual void gdcm::SimpleMemberCommand< T >::Execute (const Subject ∗ *caller,* const Event & *event*) [inline], [virtual]

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const Object

Implements gdcm::Command.

References gdcm::SimpleMemberCommand< T >::m_MemberFunction.

25.229.4.3 template<typename T > static SmartPointer<SimpleMemberCommand> gdcm::SimpleMemberCommand< T >::New ( ) [inline],[static]

Run-time type information (and related methods). Method for creation through the object factory.

References gdcm::SimpleMemberCommand< T >::SimpleMemberCommand().

25.229.4.4 template<typename T > void gdcm::SimpleMemberCommand< T >::SetCallbackFunction (T * *object,* TMemberFunctionPointer *memberFunction*) [inline]

Specify the callback function.

References gdcm::SimpleMemberCommand< T >::m_MemberFunction, and gdcm::SimpleMemberCommand< T >::m_This.

25.229.5 Member Data Documentation

25.229.5.1 template<typename T > TMemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction [protected]

Referenced by gdcm::SimpleMemberCommand< T >::Execute(), and gdcm::SimpleMemberCommand< T >::SetCallbackFunction().

25.229.5.2 template<typename T > T* gdcm::SimpleMemberCommand< T >::m_This [protected]

Referenced by gdcm::SimpleMemberCommand< T >::SetCallbackFunction().

The documentation for this class was generated from the following file:

- gdcmCommand.h

25.230 gdcm::SimpleSubjectWatcher Class Reference

SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events.

```
#include <gdcmSimpleSubjectWatcher.h>
```

Public Member Functions

- SimpleSubjectWatcher (Subject *s, const char *comment="")
- virtual ~SimpleSubjectWatcher ()

Protected Member Functions

- virtual void EndFilter ()
- virtual void ShowAbort ()
- virtual void ShowAnonymization (Subject *caller, const Event &evt)

- virtual void ShowData (Subject ∗caller, const Event &evt)
- virtual void ShowDataSet (Subject ∗caller, const Event &evt)
- virtual void ShowIteration ()
- virtual void ShowProgress (Subject ∗caller, const Event &evt)
- virtual void StartFilter ()
- void TestAbortOff ()
- void TestAbortOn ()

25.230.1 Detailed Description

SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events.

25.230.2 Constructor & Destructor Documentation

25.230.2.1 gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (Subject ∗ *s*, const char ∗ *comment* = " ")

25.230.2.2 virtual gdcm::SimpleSubjectWatcher::∼SimpleSubjectWatcher ( ) [virtual]

25.230.3 Member Function Documentation

25.230.3.1 virtual void gdcm::SimpleSubjectWatcher::EndFilter ( ) [protected], [virtual]

25.230.3.2 virtual void gdcm::SimpleSubjectWatcher::ShowAbort ( ) [protected], [virtual]

25.230.3.3 virtual void gdcm::SimpleSubjectWatcher::ShowAnonymization (Subject ∗ *caller*, const Event & *evt*) [protected], [virtual]

25.230.3.4 virtual void gdcm::SimpleSubjectWatcher::ShowData (Subject ∗ *caller*, const Event & *evt*) [protected], [virtual]

25.230.3.5 virtual void gdcm::SimpleSubjectWatcher::ShowDataSet (Subject ∗ *caller*, const Event & *evt*) [protected], [virtual]

25.230.3.6 virtual void gdcm::SimpleSubjectWatcher::ShowIteration ( ) [protected], [virtual]

25.230.3.7 virtual void gdcm::SimpleSubjectWatcher::ShowProgress (Subject ∗ *caller*, const Event & *evt*) [protected], [virtual]

25.230.3.8 virtual void gdcm::SimpleSubjectWatcher::StartFilter ( ) [protected], [virtual]

25.230.3.9 void gdcm::SimpleSubjectWatcher::TestAbortOff ( ) [protected]

25.230.3.10 void gdcm::SimpleSubjectWatcher::TestAbortOn ( ) [protected]

The documentation for this class was generated from the following file:

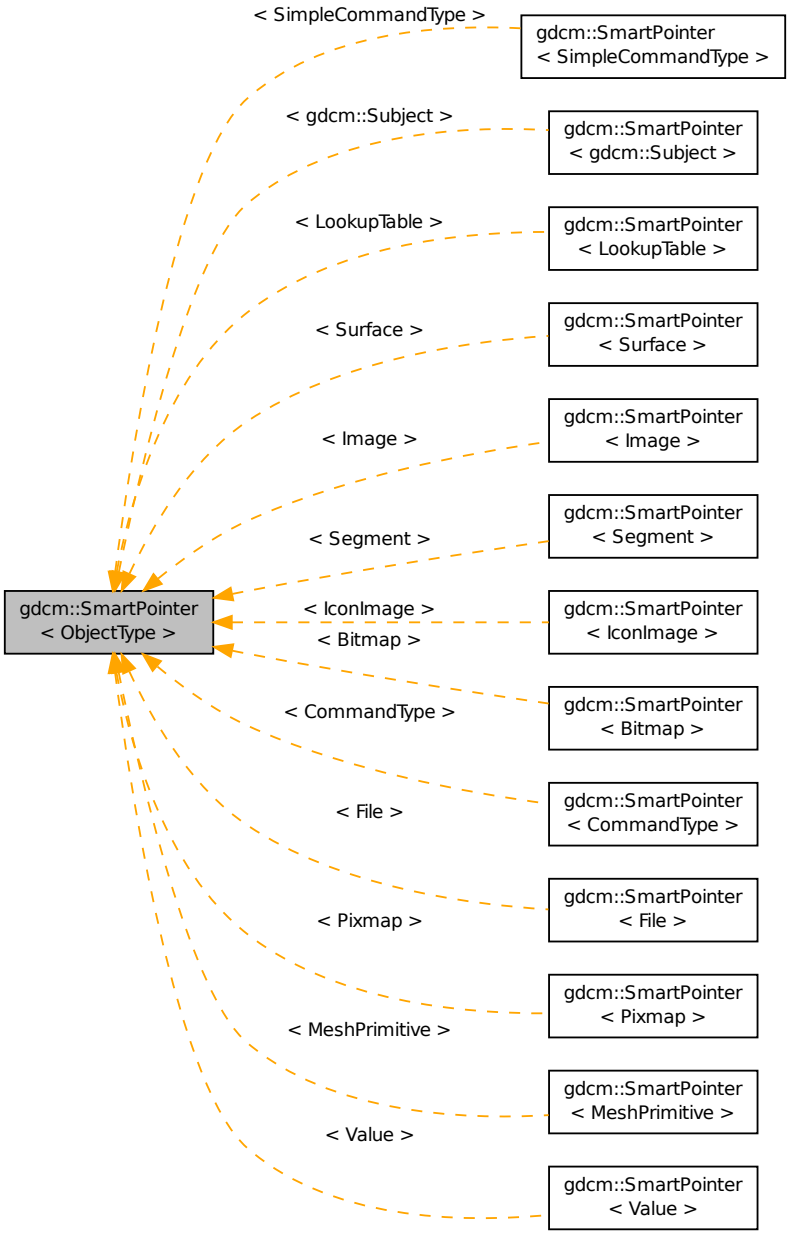
- gdcmSimpleSubjectWatcher.h

25.231 gdcm::SmartPointer< ObjectType > Class Template Reference

Class for Smart Pointer.

`#include <gdcmSmartPointer.h>`

Inheritance diagram for gdcm::SmartPointer< ObjectType >:



Public Member Functions

- SmartPointer ()
- SmartPointer (const SmartPointer< ObjectType > &p)
- SmartPointer (ObjectType *p)
- SmartPointer (ObjectType const &p)
- ~SmartPointer ()
- ObjectType * GetPointer () const

  Explicit function to retrieve the pointer.
- operator ObjectType * () const

  Return pointer to object.
- ObjectType & operator* () const
- ObjectType * operator-> () const

  Overload operator ->
- SmartPointer & operator= (SmartPointer const &r)

  Overload operator assignment.
- SmartPointer & operator= (ObjectType *r)

  Overload operator assignment.
- SmartPointer & operator= (ObjectType const &r)

25.231.1 Detailed Description

template<class ObjectType>class gdcm::SmartPointer< ObjectType >

Class for Smart Pointer.

Will only work for subclass of gdcm::Object See tr1/shared_ptr for a more general approach (not invasive) #include <tr1/memory> { shared_ptr<Bla> b(new Bla); }

Note

Class partly based on post by Bill Hubauer: `http://groups.google.com/group/comp.lang.-c++/msg/173ddc38a827a930`

See also

`http://www.davethehat.com/articles/smartp.htm`

and itk::SmartPointer

Examples:

ChangeSequenceUltrasound.cxx, CStoreQtProgress.cxx, DumpGEMSMovieGroup.cxx, Extracting_All_Resolution.-cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, Gen-AllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequence-Data.cxx, LargeVRDSExplicit.cxx, ReadAndDumpDICOMDIR.cxx, and ReadExplicitLengthSQIVR.cxx.

25.231.2 Constructor & Destructor Documentation

25.231.2.1 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer ( ) [inline]

25.231.2.2 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer (const SmartPointer< ObjectType > & *p*) [inline]

25.231.2.3 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer (ObjectType * *p*) [inline]

25.231.2.4 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer (ObjectType const & *p*) [inline]

25.231.2.5 template<class ObjectType> gdcm::SmartPointer< ObjectType >::~SmartPointer ( ) [inline]

25.231.3 Member Function Documentation

25.231.3.1 template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::GetPointer ( ) const [inline]

Explicit function to retrieve the pointer.

25.231.3.2 template<class ObjectType> gdcm::SmartPointer< ObjectType >::operator ObjectType * ( ) const [inline]

Return pointer to object.

25.231.3.3 template<class ObjectType> ObjectType& gdcm::SmartPointer< ObjectType >::operator* ( ) const [inline]

25.231.3.4 template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::operator-> ( ) const [inline]

Overload operator ->

25.231.3.5 template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (SmartPointer< ObjectType > const & *r*) [inline]

Overload operator assignment.

Referenced by gdcm::SmartPointer< Value >::operator=().

25.231.3.6 template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (ObjectType * *r*) [inline]

Overload operator assignment.

25.231.3.7 template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (ObjectType const & *r*) [inline]

The documentation for this class was generated from the following file:

- gdcmSmartPointer.h

25.232 gdcm::SOPClassUIDToIOD Class Reference

Class convert a class SOP Class UID into IOD.

`#include <gdcmSOPClassUIDToIOD.h>`

Public Types

- typedef const char ∗ const (SOPClassUIDToIODType)[2]

Static Public Member Functions

- static const char ∗ GetIOD (UIDs const &uid)
- static const char ∗ GetIODFromSOPClassUID (const char ∗sopclassuid)
- static unsigned int GetNumberOfSOPClassToIOD ()

  Return the number of SOP Class UID listed internally.
- static const char ∗ GetSOPClassUIDFromIOD (const char ∗iod)
- static SOPClassUIDToIODType & GetSOPClassUIDToIOD (unsigned int i)
- static SOPClassUIDToIODType ∗ GetSOPClassUIDToIODs ()

25.232.1 Detailed Description

Class convert a class SOP Class UID into IOD.

Reference PS 3.4 Table B.5-1 STANDARD SOP CLASSES

25.232.2 Member Typedef Documentation

25.232.2.1 typedef const char∗ gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]

25.232.3 Member Function Documentation

25.232.3.1 static const char∗ gdcm::SOPClassUIDToIOD::GetIOD (UIDs const & *uid*) [static]

Return the associated IOD based on a SOP Class UID uid (there is a one-to-one mapping from SOP Class UID to matching IOD)

Examples:

GenerateStandardSOPClasses.cxx.

25.232.3.2 static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (const char * *sopclassuid*) [static]

25.232.3.3 static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ( ) [static]

Return the number of SOP Class UID listed internally.

25.232.3.4 static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (const char * *iod*) [static]

25.232.3.5 static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (unsigned int *i*) [static]

25.232.3.6 static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ( ) [static]

The documentation for this class was generated from the following file:

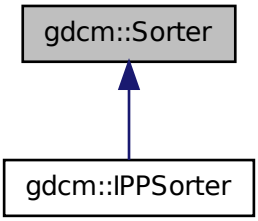
- gdcmSOPClassUIDToIOD.h

25.233 gdcm::Sorter Class Reference

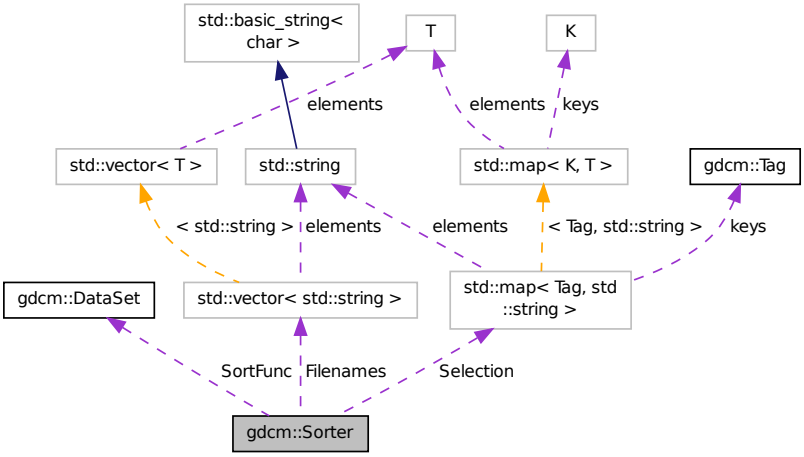
Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction.

`#include <gdcmSorter.h>`

Inheritance diagram for gdcm::Sorter:



Collaboration diagram for gdcm::Sorter:



Public Types

- typedef bool(∗ SortFunction)(DataSet const &, DataSet const &)

  Set the sort function which compares one dataset to the other.

Public Member Functions

- Sorter ()
- virtual ∼Sorter ()
- bool AddSelect (Tag const &tag, const char ∗value)

  UNSUPPORTED FOR NOW.
- const std::vector< std::string > & GetFilenames () const
- void Print (std::ostream &os) const

  Print.
- void SetSortFunction (SortFunction f)
- virtual bool Sort (std::vector< std::string > const &filenames)

  Typically the output of gdcm::Directory::GetFilenames()
- virtual bool StableSort (std::vector< std::string > const &filenames)

Protected Types

- typedef std::map< Tag,
  std::string > SelectionMap

Protected Attributes

- std::vector< std::string > Filenames
- std::map< Tag, std::string > Selection
- SortFunction SortFunc

Friends

- std::ostream & operator<< (std::ostream &_os, const Sorter &s)

25.233.1 Detailed Description

Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction.

Warning

implementation details. For now there is no cache mechanism. Which means that everytime you call Sort, all files specified as input paramater are *read*

See also

Scanner

Examples:

SortImage.cxx, and VolumeSorter.cxx.

25.233.2 Member Typedef Documentation

25.233.2.1 typedef std::map<Tag,std::string> gdcm::Sorter::SelectionMap [protected]

25.233.2.2 typedef bool(* gdcm::Sorter::SortFunction)(DataSet const &, DataSet const &)

Set the sort function which compares one dataset to the other.

25.233.3 Constructor & Destructor Documentation

25.233.3.1 gdcm::Sorter::Sorter ( )

25.233.3.2 virtual gdcm::Sorter::~Sorter ( ) [virtual]

25.233.4 Member Function Documentation

25.233.4.1 bool gdcm::Sorter::AddSelect (Tag const & *tag*, const char * *value*)

UNSUPPORTED FOR NOW.

25.233.4.2 const std::vector<std::string>& gdcm::Sorter::GetFilenames ( ) const [inline]

Return the list of filenames as sorted by the specific algorithm used. Empty by default (before Sort() is called)

Examples:

gdcmorthoplanes.cxx, reslicesphere.cxx, SortImage.cxx, and VolumeSorter.cxx.

25.233.4.3 void gdcm::Sorter::Print (std::ostream & *os*) const

Print.

Examples:

gdcmorthoplanes.cxx, SortImage.cxx, and VolumeSorter.cxx.

Referenced by gdcm::operator<<().

25.233.4.4 void gdcm::Sorter::SetSortFunction (SortFunction *f*)

Examples:

SortImage.cxx, and VolumeSorter.cxx.

25.233.4.5 virtual bool gdcm::Sorter::Sort (std::vector< std::string > const & *filenames*) [virtual]

Typically the output of gdcm::Directory::GetFilenames()

Reimplemented in gdcm::IPPSorter.

Examples:

SortImage.cxx.

25.233.4.6 virtual bool gdcm::Sorter::StableSort (std::vector< std::string > const & *filenames*) [virtual]

Examples:

SortImage.cxx, and VolumeSorter.cxx.

25.233.5 Friends And Related Function Documentation

25.233.5.1 std::ostream& operator<< (std::ostream & *_os,* const Sorter & *s*) [friend]

25.233.6 Member Data Documentation

25.233.6.1 std::vector<std::string> gdcm::Sorter::Filenames [protected]

25.233.6.2 std::map<Tag,std::string> gdcm::Sorter::Selection [protected]

25.233.6.3 SortFunction gdcm::Sorter::SortFunc [protected]

The documentation for this class was generated from the following file:

- gdcmSorter.h

25.234 gdcm::Spacing Class Reference

Class for Spacing.

`#include <gdcmSpacing.h>`

Public Types

- enum SpacingType {
  DETECTOR = 0,
  MAGNIFIED,
  CALIBRATED,
  UNKNOWN }

Public Member Functions

- Spacing ()
- ∼Spacing ()

Static Public Member Functions

- static Attribute< 0x28, 0x34 > ComputePixelAspectRatioFromPixelSpacing (const Attribute< 0x28, 0x30 > &pixelspacing)

25.234.1 Detailed Description

Class for Spacing.

It all began with a mail to WG6:

Subject: Imager Pixel Spacing vs Pixel Spacing Body: [Apologies for the duplicate post, namely to David Clunie & OFFIS team]

I have been trying to understand CP-586 in the following two cases:

On the one hand:

- DISCIMG/IMAGES/CRIMAGE taken from `http://dclunie.com/images/pixelspacingtestimages.zip`

And on the other hand:

- `http://gdcm.sourceforge.net/thingies/cr_pixelspacing.dcm`

If I understand correctly the CP, one is required to use Pixel Spacing for measurement ('true size' print) instead of Imager Pixel Spacing, since the two attributes are present and Pixel Spacing is different from Imager Pixel Spacing.

If this is correct, then the test data DISCIMG/IMAGES/CRIMAGE is incorrect. If this is incorrect (ie. I need to use Imager Pixel Spacing), then the display of cr_pixelspacing.dcm for measurement will be incorrect.

Could someone please let me know what am I missing here? I could not find any information in any header that would allow me to differentiate those.

Thank you for your time,

Ref: `http://lists.nema.org/scripts/lyris.pl?sub=488573&id=400720477` See PS 3.3-2008, Table C.7-11b IMAGE PIXEL MACRO ATTRIBUTES

Ratio of the vertical size and horizontal size of the pixels in the image specified by a pair of integer values where the first value is the vertical pixel size, and the second value is the horizontal pixel size. Required if the aspect ratio values do not have a ratio of 1:1 and the physical pixel spacing is not specified by Pixel Spacing (0028,0030), or Imager Pixel Spacing (0018,1164) or Nominal Scanned Pixel Spacing (0018,2010), either for the entire Image or per-frame in a Functional Group Macro. See C.7.6.3.1.7.

PS 3.3-2008 10.7.1.3 Pixel Spacing Value Order and Valid Values All pixel spacing related attributes shall have non-zero values, except when there is only a single row or column or pixel of data present, in which case the corresponding value may be zero.

Ref: `http://apps.sourceforge.net/mediawiki/gdcm/index.php?title=Imager_Pixel_Spacing`

25.234.2 Member Enumeration Documentation

25.234.2.1 enum gdcm::Spacing::SpacingType

Enumerator:

DETECTOR

MAGNIFIED

CALIBRATED

UNKNOWN

25.234.3 Constructor & Destructor Documentation

25.234.3.1 gdcm::Spacing::Spacing ( )

25.234.3.2 gdcm::Spacing::~Spacing ( )

25.234.4 Member Function Documentation

25.234.4.1 static Attribute<0x28,0x34> gdcm::Spacing::ComputePixelAspectRatioFromPixelSpacing (const Attribute< 0x28, 0x30 > & *pixelspacing*) [static]

The documentation for this class was generated from the following file:

- gdcmSpacing.h

25.235 gdcm::Spectroscopy Class Reference

Spectroscopy class.

```
#include <gdcmSpectroscopy.h>
```

Public Member Functions

- Spectroscopy ()

25.235.1 Detailed Description

Spectroscopy class.

25.235.2 Constructor & Destructor Documentation

25.235.2.1 gdcm::Spectroscopy::Spectroscopy ( ) [inline]

The documentation for this class was generated from the following file:

- gdcmSpectroscopy.h

25.236 gdcm::SplitMosaicFilter Class Reference

SplitMosaicFilter class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA Image Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.

```
#include <gdcmSplitMosaicFilter.h>
```

Public Member Functions

- SplitMosaicFilter ()
- ~SplitMosaicFilter ()
- bool ComputeMOSAICDimensions (unsigned int dims[3])
- File & GetFile ()
- const File & GetFile () const
- const Image & GetImage () const
- Image & GetImage ()
- void SetFile (const File &f)
- void SetImage (const Image &image)
- bool Split ()

  Split the SIEMENS MOSAIC image.

25.236.1 Detailed Description

SplitMosaicFilter class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA Image Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.

25.236.2 Constructor & Destructor Documentation

25.236.2.1 gdcm::SplitMosaicFilter::SplitMosaicFilter ( )

25.236.2.2 gdcm::SplitMosaicFilter::~SplitMosaicFilter ( )

25.236.3 Member Function Documentation

25.236.3.1 bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (unsigned int *dims[3]*)

Compute the new dimensions according to private information stored in the MOSAIC header.

25.236.3.2 File& gdcm::SplitMosaicFilter::GetFile ( ) [inline]

25.236.3.3 const File& gdcm::SplitMosaicFilter::GetFile ( ) const [inline]

25.236.3.4 const Image& gdcm::SplitMosaicFilter::GetImage ( ) const [inline]

25.236.3.5 Image& gdcm::SplitMosaicFilter::GetImage ( ) [inline]

25.236.3.6 void gdcm::SplitMosaicFilter::SetFile (const File & *f*) [inline]

25.236.3.7 void gdcm::SplitMosaicFilter::SetImage (const Image & *image*)

25.236.3.8 bool gdcm::SplitMosaicFilter::Split ( )

Split the SIEMENS MOSAIC image.

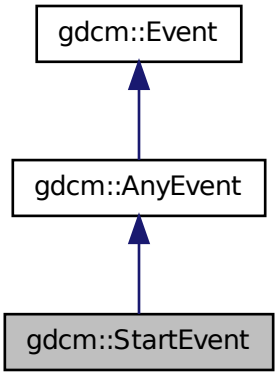
The documentation for this class was generated from the following file:

- gdcmSplitMosaicFilter.h

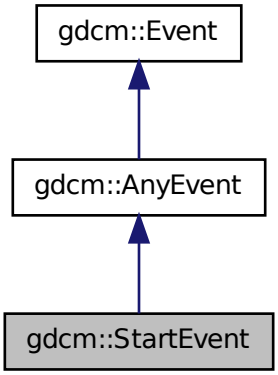
25.237 gdcm::StartEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::StartEvent:



Collaboration diagram for gdcm::StartEvent:



The documentation for this class was generated from the following file:

- gdcmEvent.h

25.238 gdcm::static_assert_test< x > Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

The documentation for this struct was generated from the following file:

- gdcmStaticAssert.h

25.239 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

Public Types

- enum { value = 1 }

25.239.1 Member Enumeration Documentation

25.239.1.1 anonymous enum

Enumerator:

value

The documentation for this struct was generated from the following file:

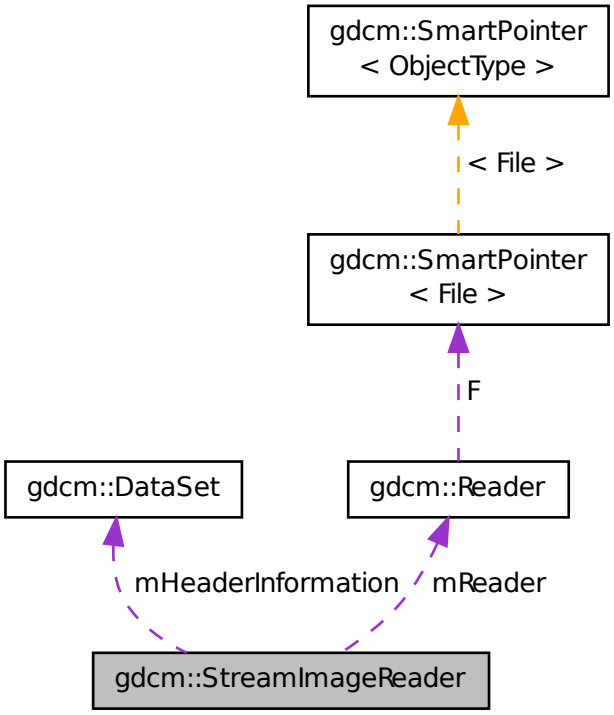
- gdcmStaticAssert.h

25.240 gdcm::StreamImageReader Class Reference

StreamImageReader.

```
#include <gdcmStreamImageReader.h>
```

Collaboration diagram for gdcm::StreamImageReader:



Public Member Functions

- StreamImageReader ()
- ∼StreamImageReader ()
- bool CanReadImage () const
- void DefinePixelExtent (uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inYMax, uint16_t inZMin=0, uint16_t inZMax=1)
- uint32_t DefineProperBufferLength () const
- std::vector< unsigned int > GetDimensionsValueForResolution (unsigned int)
- File const & GetFile () const
- bool Read (void ∗inReadBuffer, const std::size_t &inBufferLength)
- virtual bool ReadImageInformation ()
- void SetFileName (const char ∗inFileName)
- void SetStream (std::istream &inStream)

Protected Member Functions

- bool ReadImageSubregionJpegLS (char ∗inReadBuffer, const std::size_t &inBufferLength)
- virtual bool ReadImageSubregionRAW (char ∗inReadBuffer, const std::size_t &inBufferLength)

Protected Attributes

- std::streamoff mFileOffset
- std::streamoff mFileOffset1
- DataSet mHeaderInformation
- Reader mReader
- uint16_t mXMax
- uint16_t mXMin
- uint16_t mYMax
- uint16_t mYMin
- uint16_t mZMax
- uint16_t mZMin

25.240.1 Detailed Description

StreamImageReader.

Note

its role is to convert the DICOM DataSet into a gdcm::Image representation via an ITK streaming (ie, multithreaded) interface Image is different from Pixmap has it has a position and a direction in Space. Currently, this class is thread-safe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

See also

Image

Examples:

StreamImageReaderTest.cxx.

25.240.2 Constructor & Destructor Documentation

25.240.2.1 gdcm::StreamImageReader::StreamImageReader ( )

25.240.2.2 gdcm::StreamImageReader::~StreamImageReader ( )

25.240.3 Member Function Documentation

25.240.3.1 bool gdcm::StreamImageReader::CanReadImage ( ) const

Only RAW images are currently readable by the stream reader. As more streaming codecs are added, then this function will be updated to reflect those changes. Calling this function prior to reading will ensure that only streamable files are streamed. Make sure to call ReadImageInformation prior to calling this function.

Examples:

StreamImageReaderTest.cxx.

25.240.3.2 void gdcm::StreamImageReader::DefinePixelExtent (uint16_t *inXMin,* uint16_t *inXMax,* uint16_t *inYMin,* uint16_t *inYMax,* uint16_t *inZMin* = 0, uint16_t *inZMax* = 1)

Defines an image extent for the Read function. DICOM states that an image can have no more than 2^16 pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation. 15 nov 2010: added z dimension, defaults to being 1 plane large

Examples:

StreamImageReaderTest.cxx.

25.240.3.3 uint32_t gdcm::StreamImageReader::DefineProperBufferLength ( ) const

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. Call this function to determine the size of the char∗ buffer that will need to be passed in to ReadImageSubregion(). If the return is 0, then that means that the pixel extent was not defined prior

Examples:

StreamImageReaderTest.cxx.

25.240.3.4 std::vector<unsigned int> gdcm::StreamImageReader::GetDimensionsValueForResolution (unsigned *int*)

25.240.3.5 File const& gdcm::StreamImageReader::GetFile ( ) const

Returns the dataset read by ReadImageInformation Couple this with the ImageHelper to get statistics about the image, like pixel extent, to be able to initialize buffers for reading

Examples:

StreamImageReaderTest.cxx.

25.240.3.6 bool gdcm::StreamImageReader::Read (void ∗ *inReadBuffer,* const std::size_t & *inBufferLength*)

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from void∗ to std::ostream (internally) fails
3. the given buffer isn't large enough to accomodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use ImageReader instead.

Examples:

StreamImageReaderTest.cxx.

25.240.3.7 virtual bool gdcm::StreamImageReader::ReadImageInformation ( ) [virtual]

Set the spacing and dimension information for the set filename. returns false if the file is not initialized or not an image, with the pixel 0x7fe0, 0x0010 tag.

Examples:

StreamImageReaderTest.cxx.

25.240.3.8 bool gdcm::StreamImageReader::ReadImageSubregionJpegLS (char * *inReadBuffer,* const std::size_t & *inBufferLength*) [protected]

Reads the file via JpegLS. The JpegLS codec, as of this writing, requires that the entire file be read in in order to decode a subregion, so that's what's done here.

25.240.3.9 virtual bool gdcm::StreamImageReader::ReadImageSubregionRAW (char * *inReadBuffer,* const std::size_t & *inBufferLength*) [protected],[virtual]

Using the min, max, etc set by DefinePixelExtent, this will fill the given buffer Make sure to call DefinePixelExtent and to initialize the buffer with the amount given by DefineProperBufferLength prior to calling this. reads by the RAW codec; other codecs are added once implemented

25.240.3.10 void gdcm::StreamImageReader::SetFileName (const char * *inFileName*)

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal Reader class to be able to get non-pixel image information.

Examples:

StreamImageReaderTest.cxx.

25.240.3.11 void gdcm::StreamImageReader::SetStream (std::istream & *inStream*)

25.240.4 Member Data Documentation

25.240.4.1 std::streamoff gdcm::StreamImageReader::mFileOffset [protected]

25.240.4.2 std::streamoff gdcm::StreamImageReader::mFileOffset1 [protected]

25.240.4.3 DataSet gdcm::StreamImageReader::mHeaderInformation [protected]

25.240.4.4 Reader gdcm::StreamImageReader::mReader [protected]

25.240.4.5 uint16_t gdcm::StreamImageReader::mXMax [protected]

25.240.4.6 uint16_t gdcm::StreamImageReader::mXMin [protected]

25.240.4.7 uint16_t gdcm::StreamImageReader::mYMax [protected]

25.240.4.8 uint16_t gdcm::StreamImageReader::mYMin [protected]

25.240.4.9 uint16_t gdcm::StreamImageReader::mZMax [protected]

25.240.4.10 uint16_t gdcm::StreamImageReader::mZMin [protected]

The documentation for this class was generated from the following file:

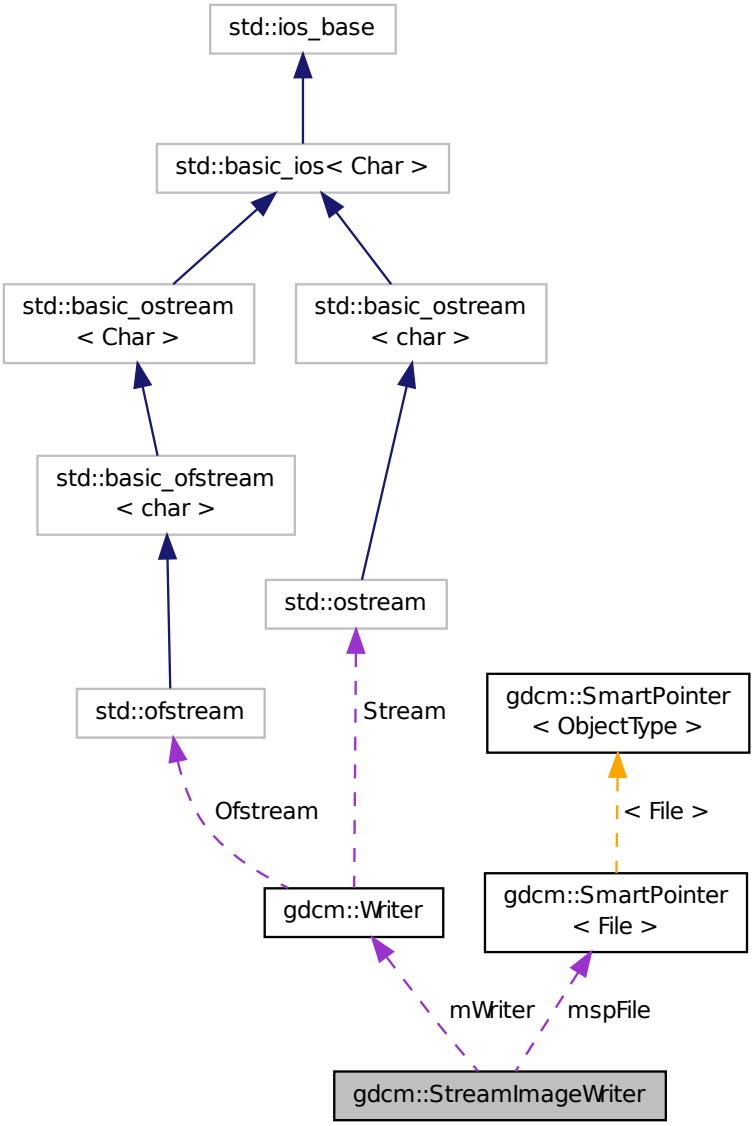
- gdcmStreamImageReader.h

25.241 gdcm::StreamImageWriter Class Reference

StreamImageReader.

```
#include <gdcmStreamImageWriter.h>
```

Collaboration diagram for gdcm::StreamImageWriter:



Public Member Functions

- StreamImageWriter ()
- ~StreamImageWriter ()
- bool CanWriteFile () const
- void DefinePixelExtent (uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inYMax, uint16_t inZMin=0, uint16_t inZMax=1)
- uint32_t DefineProperBufferLength ()

- void SetFile (const File &inFile)
- void SetFileName (const char ∗inFileName)
- void SetStream (std::ostream &inStream)
- bool Write (void ∗inWriteBuffer, const std::size_t &inBufferLength)
- virtual bool WriteImageInformation ()

Protected Member Functions

- virtual bool WriteImageSubregionRAW (char ∗inWriteBuffer, const std::size_t &inBufferLength)
- int WriteRawHeader (RAWCodec ∗inCodec, std::ostream ∗inStream)

Protected Attributes

- int mElementOffsets
- int mElementOffsets1
- SmartPointer< File > mspFile
- Writer mWriter
- uint16_t mXMax
- uint16_t mXMin
- uint16_t mYMax
- uint16_t mYMin
- uint16_t mZMax
- uint16_t mZMin

25.241.1 Detailed Description

StreamImageReader.

Note

its role is to convert the DICOM DataSet into a gdcm::Image representation via an ITK streaming (ie, multithreaded) interface Image is different from Pixmap has it has a position and a direction in Space. Currently, this class is thread-safe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

See also

Image

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.241.2 Constructor & Destructor Documentation

25.241.2.1 gdcm::StreamImageWriter::StreamImageWriter ( )

25.241.2.2 gdcm::StreamImageWriter::∼StreamImageWriter ( )

25.241.3 Member Function Documentation

25.241.3.1 bool gdcm::StreamImageWriter::CanWriteFile ( ) const

This function determines if a file can even be written using the streaming writer unlike the reader, can be called before WriteImageInformation, but must be called after SetFile.

Examples:

Extracting_All_Resolution.cxx, and Fake_Image_Using_Stream_Image_Writer.cxx.

25.241.3.2 void gdcm::StreamImageWriter::DefinePixelExtent (uint16_t *inXMin,* uint16_t *inXMax,* uint16_t *inYMin,* uint16_t *inYMax,* uint16_t *inZMin* = 0, uint16_t *inZMax* = 1)

Defines an image extent for the Read function. DICOM states that an image can have no more than 2^16 pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation. 15 nov 2010: added z dimension, defaults to being 1 plane large

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.241.3.3 uint32_t gdcm::StreamImageWriter::DefineProperBufferLength ( )

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. If the return is 0, then that means that the pixel extent was not defined prior this return is for RAW inputs which are then encoded by the writer, but are used to ensure that the writer gets the proper buffer size

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.241.3.4 void gdcm::StreamImageWriter::SetFile (const File & *inFile*)

Set the image information to be written to disk that is everything but the pixel information: (7fe0,0010) PixelData

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.241.3.5 void gdcm::StreamImageWriter::SetFileName (const char * *inFileName*)

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal Reader class to be able to get non-pixel image information.

25.241.3.6 void gdcm::StreamImageWriter::SetStream (std::ostream & *inStream*)

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.241.3.7 bool gdcm::StreamImageWriter::Write (void * *inWriteBuffer,* const std::size_t & *inBufferLength*)

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from void* to std::ostream (internally) fails
3. the given buffer isn't large enough to accomodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use ImageReader instead.

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.241.3.8 virtual bool gdcm::StreamImageWriter::WriteImageInformation ( ) [virtual]

Write the header information to disk, and a bunch of zeros for the actual pixel information Of course, if we're doing a non-compressed format, that works but if it's compressed, we have to force the ordering of chunks that are written.

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.241.3.9 virtual bool gdcm::StreamImageWriter::WriteImageSubregionRAW (char * *inWriteBuffer,* const std::size_t & *inBufferLength*) [protected],[virtual]

Using the min, max, etc set by DefinePixelExtent, this will fill the given buffer Make sure to call DefinePixelExtent and to initialize the buffer with the amount given by DefineProperBufferLength prior to calling this. reads by the RAW codec; other codecs are added once implemented

25.241.3.10 int gdcm::StreamImageWriter::WriteRawHeader (RAWCodec * *inCodec,* std::ostream * *inStream*) [protected]

when writing a raw file, we know the full extent, and can just write the first 12 bytes out (the tag, the VR, and the size) when we do compressed files, we'll do it in chunks, as described in 2009-3, part 5, Annex A, section 4. Pass the raw codec so that in the rare case of a bigendian explicit raw, the first 12 bytes written out should still be kosher. returns -1 if there's any failure, or the complete offset (12 bytes) if it works. Those 12 bytes are then added to the position in order to determine where to write.

25.241.4 Member Data Documentation

25.241.4.1 int gdcm::StreamImageWriter::mElementOffsets [protected]

The result of WriteRawHeader (or another header, when that's implemented) This result is saved so that the first N bytes aren't constantly being rewritten for each chunk that's passed in. For compressed data, the offset table will require rewrites of data.

25.241.4.2 int gdcm::StreamImageWriter::mElementOffsets1 [protected]

25.241.4.3 SmartPointer<File> gdcm::StreamImageWriter::mspFile [protected]

25.241.4.4 Writer gdcm::StreamImageWriter::mWriter [protected]

25.241.4.5 uint16_t gdcm::StreamImageWriter::mXMax [protected]

25.241.4.6 uint16_t gdcm::StreamImageWriter::mXMin [protected]

25.241.4.7 uint16_t gdcm::StreamImageWriter::mYMax [protected]

25.241.4.8 uint16_t gdcm::StreamImageWriter::mYMin [protected]

25.241.4.9 uint16_t gdcm::StreamImageWriter::mZMax [protected]

25.241.4.10 uint16_t gdcm::StreamImageWriter::mZMin [protected]

The documentation for this class was generated from the following file:

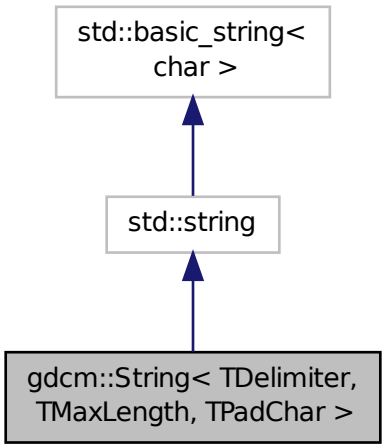
- gdcmStreamImageWriter.h

25.242 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference

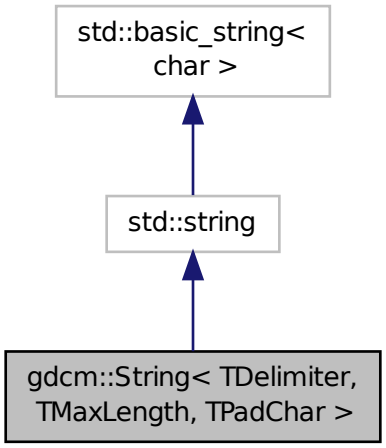
String.

`#include <gdcmString.h>`

Inheritance diagram for gdcm::String< TDelimiter, TMaxLength, TPadChar >:



Collaboration diagram for gdcm::String< TDelimiter, TMaxLength, TPadChar >:



Public Types

- typedef std::string::const_iterator const_iterator
- typedef std::string::const_reference const_reference
- typedef std::string::const_reverse_iterator const_reverse_iterator
- typedef std::string::difference_type difference_type
- typedef std::string::iterator iterator
- typedef std::string::pointer pointer
- typedef std::string::reference reference
- typedef std::string::reverse_iterator reverse_iterator
- typedef std::string::size_type size_type
- typedef std::string::value_type value_type

Public Member Functions

- String ()

  String constructors.
- String (const value_type ∗s)
- String (const value_type ∗s, size_type n)
- String (const std::string &s, size_type pos=0, size_type n=npos)
- bool IsValid () const

  return if string is valid

- operator const char ∗ () const

  WARNING: Trailing \0 might be lost in this operation:
- std::string Trim () const
- gdcm::String< TDelimiter, TMaxLength, TPadChar > Truncate () const

25.242.1 Detailed Description

template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>class gdcm::String< TDelimiter, TMaxLength, TPadChar >

String.

Note

TDelimiter template parameter is used to separate multiple String (VM1 >) TMaxLength is only a hint. Noone actually respect the max length TPadChar is the string padding (0 or space)

25.242.2 Member Typedef Documentation

25.242.2.1 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_iterator **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::const_iterator

25.242.2.2 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reference **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::**const_reference**

25.242.2.3 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reverse_iterator **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::const_reverse_iterator

25.242.2.4 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::difference_type **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::**difference_type**

25.242.2.5 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::iterator **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::iterator

25.242.2.6 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::pointer **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::**pointer**

25.242.2.7 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reference **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::**reference**

25.242.2.8 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reverse_iterator **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::reverse_iterator

25.242.2.9 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::size_type **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::**size_type**

25.242.2.10 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::value_type **gdcm::String**< TDelimiter, TMaxLength, TPadChar >::**value_type**

25.242.3 Constructor & Destructor Documentation

25.242.3.1 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( ) [inline]

String constructors.

25.242.3.2 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * *s*) [inline]

25.242.3.3 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * *s*, size_type *n*) [inline]

25.242.3.4 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const std::string & *s*, size_type *pos* = 0, size_type *n* = npos) [inline]

25.242.4 Member Function Documentation

25.242.4.1 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> bool gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid ( ) const [inline]

return if string is valid

Referenced by gdcm::LO::IsValid(), and gdcm::String< TDelimiter, TMaxLength, TPadChar >::Truncate().

25.242.4.2 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::operator const char * ( ) const [inline]

WARNING: Trailing \0 might be lost in this operation:

25.242.4.3 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim ( ) const [inline]

Trim function is required to return a std::string object, otherwise we could not create a gdcm::String object with an odd number of bytes...

25.242.4.4 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String<TDelimiter, TMaxLength, TPadChar> gdcm::String< TDelimiter, TMaxLength, TPadChar >::Truncate ( ) const [inline]

References gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid().

The documentation for this class was generated from the following file:

- gdcmString.h

25.243 gdcm::StringFilter Class Reference

StringFilter StringFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

```
#include <gdcmStringFilter.h>
```

Public Member Functions

- StringFilter ()
- ∼StringFilter ()
- bool ExecuteQuery (std::string const &query, std::string &value) const
- std::string FromString (const Tag &t, const char ∗value, VL const &vl)

  DEPRECATED: NEVER USE IT.
- std::string FromString (const Tag &t, const char ∗value, size_t len)
- File & GetFile ()
- const File & GetFile () const
- void SetDicts (const Dicts &dicts)

  Allow user to pass in there own dicts.
- void SetFile (const File &f)

  Set/Get File.
- std::string ToString (const Tag &t) const

  Convert to string the ByteValue contained in a DataElement.
- std::pair< std::string,
  std::string > ToStringPair (const Tag &t) const
- void UseDictAlways (bool)

Protected Member Functions

- bool ExecuteQuery (std::string const &query, DataSet const &ds, std::string &value) const
- std::pair< std::string,
  std::string > ToStringPair (const Tag &t, DataSet const &ds) const

25.243.1 Detailed Description

StringFilter StringFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

Examples:

ReadAndPrintAttributes.cxx.

25.243.2 Constructor & Destructor Documentation

25.243.2.1 gdcm::StringFilter::StringFilter ( )

25.243.2.2 gdcm::StringFilter::∼StringFilter ( )

25.243.3 Member Function Documentation

25.243.3.1 bool gdcm::StringFilter::ExecuteQuery (std::string const & *query*, std::string & *value*) const

Execute the XPATH query to find a value (as string) return false when attribute is not found (or an error in the XPATH query) You need to make sure that your XPATH query is syntatically correct

25.243.3.2 bool gdcm::StringFilter::ExecuteQuery (std::string const & *query,* DataSet const & *ds,* std::string & *value*) const [protected]

25.243.3.3 std::string gdcm::StringFilter::FromString (const Tag & *t,* const char * *value,* VL const & *vl*)

DEPRECATED: NEVER USE IT.

25.243.3.4 std::string gdcm::StringFilter::FromString (const Tag & *t,* const char * *value,* size_t *len*)

25.243.3.5 File& gdcm::StringFilter::GetFile ( ) [inline]

25.243.3.6 const File& gdcm::StringFilter::GetFile ( ) const [inline]

25.243.3.7 void gdcm::StringFilter::SetDicts (const Dicts & *dicts*)

Allow user to pass in there own dicts.

25.243.3.8 void gdcm::StringFilter::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

ReadAndPrintAttributes.cxx.

25.243.3.9 std::string gdcm::StringFilter::ToString (const Tag & *t*) const

Convert to string the ByteValue contained in a DataElement.

Examples:

ReadAndPrintAttributes.cxx.

25.243.3.10 std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & *t*) const

Convert to string the ByteValue contained in a DataElement the returned elements are: pair.first : the name as found in the dictionary of DataElement pari.second : the value encoded into a string (US,UL...) are properly converted

Examples:

ReadAndPrintAttributes.cxx.

25.243.3.11 std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & *t,* DataSet const & *ds*) const [protected]

25.243.3.12 void gdcm::StringFilter::UseDictAlways (bool) [inline]

The documentation for this class was generated from the following file:

- gdcmStringFilter.h

25.244 gdcm::Study Class Reference

Study.

`#include <gdcmStudy.h>`

Public Member Functions

- Study ()

25.244.1 Detailed Description

Study.

25.244.2 Constructor & Destructor Documentation

25.244.2.1 gdcm::Study::Study ( ) [inline]

The documentation for this class was generated from the following file:

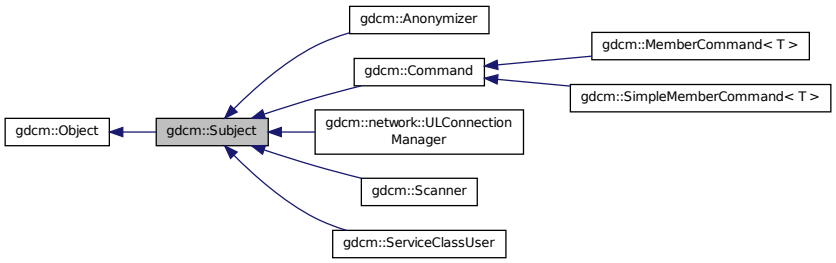
- gdcmStudy.h

25.245 gdcm::Subject Class Reference

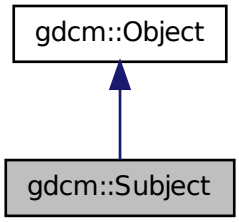
Subject.

`#include <gdcmSubject.h>`

Inheritance diagram for gdcm::Subject:



Collaboration diagram for gdcm::Subject:



Public Member Functions

- Subject ()
- ∼Subject ()
- unsigned long AddObserver (const Event &event, Command ∗)
- unsigned long AddObserver (const Event &event, Command ∗) const
- Command ∗ GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Additional Inherited Members

25.245.1 Detailed Description

Subject.

See also

Command Event

25.245.2 Constructor & Destructor Documentation

25.245.2.1 gdcm::Subject::Subject ( )

25.245.2.2 gdcm::Subject::∼Subject ( )

25.245.3 Member Function Documentation

25.245.3.1 unsigned long gdcm::Subject::AddObserver (const Event & *event,* Command *)

Allow people to add/remove/invoke observers (callbacks) to any GDCM object. This is an implementation of the subject/observer design pattern. An observer is added by specifying an event to respond to and an gdcm::Command to execute. It returns an unsigned long tag which can be used later to remove the event or retrieve the command. The memory for the Command becomes the responsibility of this object, so don't pass the same instance of a command to two different objects

25.245.3.2 unsigned long gdcm::Subject::AddObserver (const Event & *event,* Command *) const

25.245.3.3 Command* gdcm::Subject::GetCommand (unsigned long *tag*)

Get the command associated with the given tag. NOTE: This returns a pointer to a Command, but it is safe to asign this to a Command::Pointer. Since Command inherits from LightObject, at this point in the code, only a pointer or a reference to the Command can be used.

25.245.3.4 bool gdcm::Subject::HasObserver (const Event & *event*) const

Return true if an observer is registered for this event.

25.245.3.5 void gdcm::Subject::InvokeEvent (const Event &)

Call Execute on all the Commands observing this event id.

25.245.3.6 void gdcm::Subject::InvokeEvent (const Event &) const

Call Execute on all the Commands observing this event id. The actions triggered by this call doesn't modify this object.

25.245.3.7 void gdcm::Subject::RemoveAllObservers ( )

Remove all observers .

25.245.3.8 void gdcm::Subject::RemoveObserver (unsigned long *tag*)

Remove the observer with this tag value.

The documentation for this class was generated from the following file:

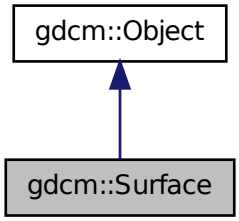
- gdcmSubject.h

25.246 gdcm::Surface Class Reference

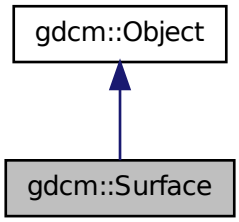
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

```
#include <gdcmSurface.h>
```

Inheritance diagram for gdcm::Surface:



Collaboration diagram for gdcm::Surface:



Public Types

- enum STATES {
  NO = 0,
  YES,
  UNKNOWN,
  STATES_END }
- enum VIEWType {
  SURFACE = 0,
  WIREFRAME,
  POINTS,
  VIEWType_END }

  Enumeration for Recommended Presentation Type.

Public Member Functions

- Surface ()

- virtual ∼Surface ()
- SegmentHelper::BasicCodedEntry const & GetAlgorithmFamily () const
- SegmentHelper::BasicCodedEntry & GetAlgorithmFamily ()
- const char ∗ GetAlgorithmName () const
- const char ∗ GetAlgorithmVersion () const
- const float ∗ GetAxisOfRotation () const
- const float ∗ GetCenterOfRotation () const
- STATES GetFiniteVolume () const
- STATES GetManifold () const
- float GetMaximumPointDistance () const
- float GetMeanPointDistance () const
- MeshPrimitive const & GetMeshPrimitive () const
- MeshPrimitive & GetMeshPrimitive ()
- unsigned long GetNumberOfSurfacePoints () const
- unsigned long GetNumberOfVectors () const
- const DataElement & GetPointCoordinatesData () const
- DataElement & GetPointCoordinatesData ()
- const float ∗ GetPointPositionAccuracy () const
- const float ∗ GetPointsBoundingBoxCoordinates () const
- SegmentHelper::BasicCodedEntry const & GetProcessingAlgorithm () const
- SegmentHelper::BasicCodedEntry & GetProcessingAlgorithm ()
- const unsigned short ∗ GetRecommendedDisplayCIELabValue () const
- unsigned short GetRecommendedDisplayCIELabValue (const unsigned int idx) const
- unsigned short GetRecommendedDisplayGrayscaleValue () const
- float GetRecommendedPresentationOpacity () const
- VIEWType GetRecommendedPresentationType () const
- const char ∗ GetSurfaceComments () const
- unsigned long GetSurfaceNumber () const
- bool GetSurfaceProcessing () const
- const char ∗ GetSurfaceProcessingDescription () const
- float GetSurfaceProcessingRatio () const
- const float ∗ GetVectorAccuracy () const
- const DataElement & GetVectorCoordinateData () const
- DataElement & GetVectorCoordinateData ()
- unsigned short GetVectorDimensionality () const
- void SetAlgorithmFamily (SegmentHelper::BasicCodedEntry const &BSE)
- void SetAlgorithmName (const char ∗str)
- void SetAlgorithmVersion (const char ∗str)
- void SetAxisOfRotation (const float ∗axis)
- void SetCenterOfRotation (const float ∗center)
- void SetFiniteVolume (STATES state)
- void SetManifold (STATES state)
- void SetMaximumPointDistance (float maximum)
- void SetMeanPointDistance (float average)
- void SetMeshPrimitive (MeshPrimitive &mp)
- void SetNumberOfSurfacePoints (const unsigned long nb)
- void SetNumberOfVectors (const unsigned long nb)
- void SetPointCoordinatesData (DataElement const &de)
- void SetPointPositionAccuracy (const float ∗accuracies)

- void SetPointsBoundingBoxCoordinates (const float ∗coordinates)
- void SetProcessingAlgorithm (SegmentHelper::BasicCodedEntry const &BSE)
- void SetRecommendedDisplayCIELabValue (const unsigned short vl[3])
- void SetRecommendedDisplayCIELabValue (const unsigned short vl, const unsigned int idx=0)
- void SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > &vl)
- void SetRecommendedDisplayGrayscaleValue (const unsigned short vl)
- void SetRecommendedPresentationOpacity (const float opacity)
- void SetRecommendedPresentationType (VIEWType type)
- void SetSurfaceComments (const char ∗comment)
- void SetSurfaceNumber (const unsigned long nb)
- void SetSurfaceProcessing (bool b)
- void SetSurfaceProcessingDescription (const char ∗description)
- void SetSurfaceProcessingRatio (const float ratio)
- void SetVectorAccuracy (const float ∗accuracy)
- void SetVectorCoordinateData (DataElement const &de)
- void SetVectorDimensionality (const unsigned short dim)

Static Public Member Functions

- static STATES GetSTATES (const char ∗state)
- static const char ∗ GetSTATESString (STATES state)
- static VIEWType GetVIEWType (const char ∗type)
- static const char ∗ GetVIEWTypeString (VIEWType type)

Additional Inherited Members

25.246.1 Detailed Description

This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

25.246.2 Member Enumeration Documentation

25.246.2.1 enum gdcm::Surface::STATES

Enumerator:

NO

YES

UNKNOWN

STATES_END

25.246.2.2 enum gdcm::Surface::VIEWType

Enumeration for Recommended Presentation Type.

See also

Tag(0x0066, 0x000D) and PS 3.3 C.27.1.1.3

Enumerator:

SURFACE

WIREFRAME

POINTS

VIEWType_END

25.246.3 Constructor & Destructor Documentation

25.246.3.1 gdcm::Surface::Surface ( )

25.246.3.2 virtual gdcm::Surface::~Surface ( ) [virtual]

25.246.4 Member Function Documentation

25.246.4.1 SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetAlgorithmFamily ( ) const

25.246.4.2 SegmentHelper::BasicCodedEntry& gdcm::Surface::GetAlgorithmFamily ( )

25.246.4.3 const char* gdcm::Surface::GetAlgorithmName ( ) const

25.246.4.4 const char* gdcm::Surface::GetAlgorithmVersion ( ) const

25.246.4.5 const float* gdcm::Surface::GetAxisOfRotation ( ) const

Note

Pointer is null if undefined

25.246.4.6 const float* gdcm::Surface::GetCenterOfRotation ( ) const

Note

Pointer is null if undefined

25.246.4.7 STATES gdcm::Surface::GetFiniteVolume ( ) const

25.246.4.8 STATES gdcm::Surface::GetManifold ( ) const

25.246.4.9 float gdcm::Surface::GetMaximumPointDistance ( ) const

25.246.4.10 float gdcm::Surface::GetMeanPointDistance ( ) const

25.246.4.11 **MeshPrimitive** const& gdcm::Surface::GetMeshPrimitive ( ) const

25.246.4.12 **MeshPrimitive**& gdcm::Surface::GetMeshPrimitive ( )

25.246.4.13 unsigned long gdcm::Surface::GetNumberOfSurfacePoints ( ) const

25.246.4.14 unsigned long gdcm::Surface::GetNumberOfVectors ( ) const

25.246.4.15 const **DataElement**& gdcm::Surface::GetPointCoordinatesData ( ) const

25.246.4.16 **DataElement**& gdcm::Surface::GetPointCoordinatesData ( )

25.246.4.17 const float* gdcm::Surface::GetPointPositionAccuracy ( ) const

Note

Pointer is null if undefined

25.246.4.18 const float* gdcm::Surface::GetPointsBoundingBoxCoordinates ( ) const

Note

Pointer is null if undefined

25.246.4.19 **SegmentHelper::BasicCodedEntry** const& gdcm::Surface::GetProcessingAlgorithm ( ) const

25.246.4.20 **SegmentHelper::BasicCodedEntry**& gdcm::Surface::GetProcessingAlgorithm ( )

25.246.4.21 const unsigned short* gdcm::Surface::GetRecommendedDisplayCIELabValue ( ) const

25.246.4.22 unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (const unsigned int *idx*) const

25.246.4.23 unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue ( ) const

25.246.4.24 float gdcm::Surface::GetRecommendedPresentationOpacity ( ) const

25.246.4.25 **VIEWType** gdcm::Surface::GetRecommendedPresentationType ( ) const

25.246.4.26 static **STATES** gdcm::Surface::GetSTATES (const char * *state*) [static]

25.246.4.27 static const char* gdcm::Surface::GetSTATESString (**STATES** *state*) [static]

25.246.4.28 const char* gdcm::Surface::GetSurfaceComments ( ) const

25.246.4.29 unsigned long gdcm::Surface::GetSurfaceNumber ( ) const

25.246.4.30 bool gdcm::Surface::GetSurfaceProcessing ( ) const

25.246.4.31 const char* gdcm::Surface::GetSurfaceProcessingDescription ( ) const

25.246.4.32 float gdcm::Surface::GetSurfaceProcessingRatio ( ) const

25.246.4.33 const float* gdcm::Surface::GetVectorAccuracy ( ) const

25.246.4.34 const **DataElement**& gdcm::Surface::GetVectorCoordinateData ( ) const

25.246.4.35 **DataElement**& gdcm::Surface::GetVectorCoordinateData ( )

25.246.4.36 unsigned short gdcm::Surface::GetVectorDimensionality ( ) const

25.246.4.37 static **VIEWType** gdcm::Surface::GetVIEWType (const char * *type*) [static]

25.246.4.38 static const char* gdcm::Surface::GetVIEWTypeString (**VIEWType** *type*) [static]

25.246.4.39 void gdcm::Surface::SetAlgorithmFamily (**SegmentHelper::BasicCodedEntry** const & *BSE*)

25.246.4.40 void gdcm::Surface::SetAlgorithmName (const char * *str*)

25.246.4.41 void gdcm::Surface::SetAlgorithmVersion (const char * *str*)

25.246.4.42 void gdcm::Surface::SetAxisOfRotation (const float * *axis*)

25.246.4.43 void gdcm::Surface::SetCenterOfRotation (const float * *center*)

25.246.4.44 void gdcm::Surface::SetFiniteVolume (**STATES** *state*)

25.246.4.45 void gdcm::Surface::SetManifold (**STATES** *state*)

25.246.4.46 void gdcm::Surface::SetMaximumPointDistance (float *maximum*)

25.246.4.47 void gdcm::Surface::SetMeanPointDistance (float *average*)

25.246.4.48 void gdcm::Surface::SetMeshPrimitive (**MeshPrimitive** & *mp*)

25.246.4.49 void gdcm::Surface::SetNumberOfSurfacePoints (const unsigned long *nb*)

25.246.4.50 void gdcm::Surface::SetNumberOfVectors (const unsigned long *nb*)

25.246.4.51 void gdcm::Surface::SetPointCoordinatesData (**DataElement** const & *de*)

25.246.4.52 void gdcm::Surface::SetPointPositionAccuracy (const float * *accuracies*)

25.246.4.53 void gdcm::Surface::SetPointsBoundingBoxCoordinates (const float * *coordinates*)

25.246.4.54 void gdcm::Surface::SetProcessingAlgorithm (**SegmentHelper::BasicCodedEntry** const & *BSE*)

25.246.4.55 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl[3]*)

25.246.4.56 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl,* const unsigned int *idx* = 0)

25.246.4.57 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > & *vl*)

25.246.4.58 void gdcm::Surface::SetRecommendedDisplayGrayscaleValue (const unsigned short *vl*)

25.246.4.59 void gdcm::Surface::SetRecommendedPresentationOpacity (const float *opacity*)

25.246.4.60 void gdcm::Surface::SetRecommendedPresentationType (VIEWType *type*)

25.246.4.61 void gdcm::Surface::SetSurfaceComments (const char * *comment*)

25.246.4.62 void gdcm::Surface::SetSurfaceNumber (const unsigned long *nb*)

25.246.4.63 void gdcm::Surface::SetSurfaceProcessing (bool *b*)

25.246.4.64 void gdcm::Surface::SetSurfaceProcessingDescription (const char * *description*)

25.246.4.65 void gdcm::Surface::SetSurfaceProcessingRatio (const float *ratio*)

25.246.4.66 void gdcm::Surface::SetVectorAccuracy (const float * *accuracy*)

25.246.4.67 void gdcm::Surface::SetVectorCoordinateData (DataElement const & *de*)

25.246.4.68 void gdcm::Surface::SetVectorDimensionality (const unsigned short *dim*)

The documentation for this class was generated from the following file:

- gdcmSurface.h

25.247 gdcm::SurfaceHelper Class Reference

```
#include <gdcmSurfaceHelper.h>
```

Public Types

- typedef std::vector< unsigned short > ColorArray

Static Public Member Functions

- template<typename T , typename U >
  static std::vector< T > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U rangeMax=255)

  Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename U >
  static std::vector< float > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U rangeMax=255)

  Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename T , typename U >
  static ColorArray RGBToRecommendedDisplayCIELab (const std::vector< T > &RGB, const U rangeMax=255)

  Convert a RGB color into DICOM CIE-Lab (ready to write).
- template<typename T , typename U >
  static unsigned short RGBToRecommendedDisplayGrayscale (const std::vector< T > &RGB, const U rangeMax=255)

  Convert a RGB color into DICOM grayscale (ready to write).

25.247.1 Member Typedef Documentation

25.247.1.1 typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray

25.247.2 Member Function Documentation

25.247.2.1 template<typename U > std::vector< float > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (const ColorArray & *CIELab*, const U *rangeMax* = 255) [static]

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

CIELab	DICOM CIE-Lab array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of CIELab components.
U	Type of rangeMax value.

25.247.2.2 template<typename U > static std::vector<float> gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (const ColorArray & *CIELab*, const U *rangeMax* = 255) [static]

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

CIELab	DICOM CIE-Lab array.
rangeMax	Max value of the RGB range.

Template Parameters

U	Type of rangeMax value.

25.247.2.3 template<typename T , typename U > SurfaceHelper::ColorArray gdcm::SurfaceHelper::RGBToRecommendedDisplayCIELab (const std::vector< T > & *RGB*, const U *rangeMax* = 255) [static]

Convert a RGB color into DICOM CIE-Lab (ready to write).

See also

PS 3.3 C.10.7.1.1

Parameters

RGB	RGB array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of RGB components.
U	Type of rangeMax value.

25.247.2.4 template<typename T , typename U > unsigned short gdcm::SurfaceHelper::RGBToRecommendedDisplayGrayscale (const std::vector< T > & *RGB*, const U *rangeMax* = 255) [static]

Convert a RGB color into DICOM grayscale (ready to write).

See also

PS 3.3 C.27.1 tag(0062,000C)

Parameters

RGB	RGB array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of RGB components.
U	Type of rangeMax value.

The documentation for this class was generated from the following file:

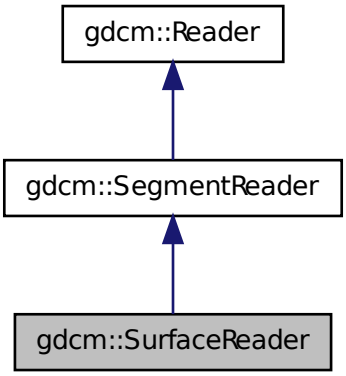
- gdcmSurfaceHelper.h

25.248 gdcm::SurfaceReader Class Reference

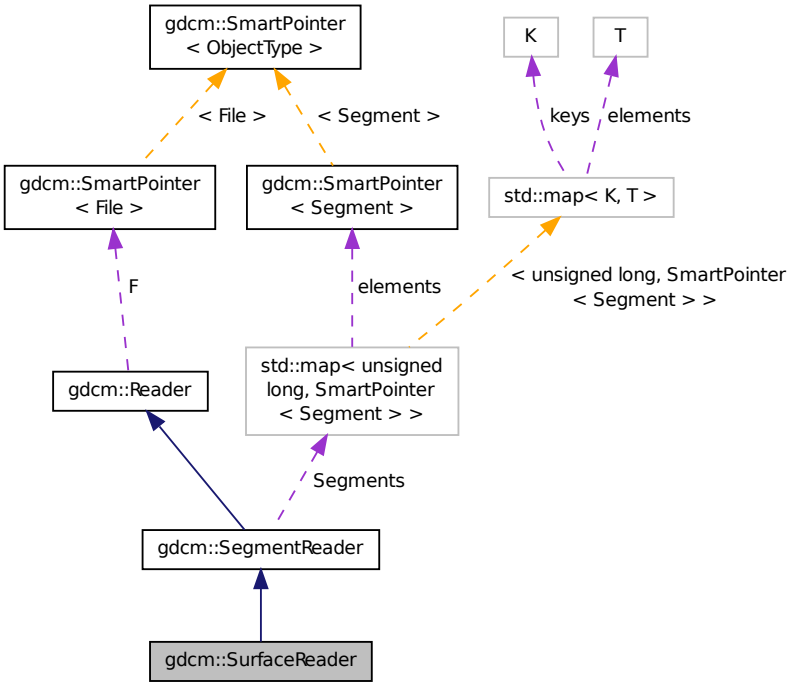
This class defines a SURFACE IE reader. It reads surface mesh module attributes.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for gdcm::SurfaceReader:



Collaboration diagram for gdcm::SurfaceReader:



Public Member Functions

- SurfaceReader ()
- virtual ~SurfaceReader ()
- unsigned long GetNumberOfSurfaces () const
- virtual bool Read ()

  Read.

Protected Member Functions

- bool ReadPointMacro (SmartPointer< Surface > surface, const DataSet &surfaceDS)
- bool ReadSurface (const Item &surfaceItem, const unsigned long idx)
- bool ReadSurfaces ()

Additional Inherited Members

25.248.1 Detailed Description

This class defines a SURFACE IE reader. It reads surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

25.248.2 Constructor & Destructor Documentation

25.248.2.1 gdcm::SurfaceReader::SurfaceReader ( )

25.248.2.2 virtual gdcm::SurfaceReader::~SurfaceReader ( ) [virtual]

25.248.3 Member Function Documentation

25.248.3.1 unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ( ) const

25.248.3.2 virtual bool gdcm::SurfaceReader::Read ( ) [virtual]

Read.

Reimplemented from gdcm::SegmentReader.

25.248.3.3 bool gdcm::SurfaceReader::ReadPointMacro (SmartPointer< Surface > *surface,* const DataSet & *surfaceDS*) [protected]

25.248.3.4 bool gdcm::SurfaceReader::ReadSurface (const Item & *surfaceItem,* const unsigned long *idx*) [protected]

25.248.3.5 bool gdcm::SurfaceReader::ReadSurfaces ( ) [protected]

The documentation for this class was generated from the following file:

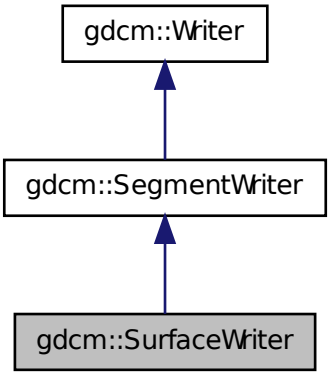
- gdcmSurfaceReader.h

25.249 gdcm::SurfaceWriter Class Reference

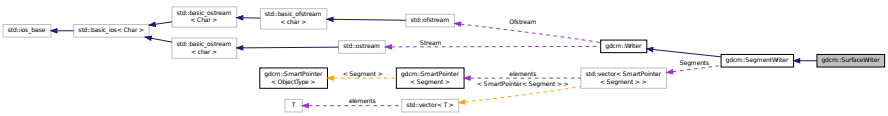
This class defines a SURFACE IE writer. It writes surface mesh module attributes.

`#include <gdcmSurfaceWriter.h>`

Inheritance diagram for gdcm::SurfaceWriter:



Collaboration diagram for gdcm::SurfaceWriter:



Public Member Functions

- SurfaceWriter ()
- virtual ~SurfaceWriter ()
- unsigned long GetNumberOfSurfaces ()
- void SetNumberOfSurfaces (const unsigned long nb)
- bool Write ()

  Write.

Protected Member Functions

- void ComputeNumberOfSurfaces ()
- bool PrepareWrite ()
- bool PrepareWritePointMacro (SmartPointer< Surface > surface, DataSet &surfaceDS, const TransferSyntax &ts)

Protected Attributes

- unsigned long NumberOfSurfaces

Additional Inherited Members

25.249.1 Detailed Description

This class defines a SURFACE IE writer. It writes surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

25.249.2 Constructor & Destructor Documentation

25.249.2.1 gdcm::SurfaceWriter::SurfaceWriter ( )

25.249.2.2 virtual gdcm::SurfaceWriter::~SurfaceWriter ( ) [virtual]

25.249.3 Member Function Documentation

25.249.3.1 void gdcm::SurfaceWriter::ComputeNumberOfSurfaces ( ) [protected]

25.249.3.2 unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ( )

25.249.3.3 bool gdcm::SurfaceWriter::PrepareWrite ( ) [protected]

Reimplemented from gdcm::SegmentWriter.

25.249.3.4 bool gdcm::SurfaceWriter::PrepareWritePointMacro (SmartPointer< Surface > *surface*, DataSet & *surfaceDS*, const TransferSyntax & *ts*) [protected]

25.249.3.5 void gdcm::SurfaceWriter::SetNumberOfSurfaces (const unsigned long *nb*)

25.249.3.6 bool gdcm::SurfaceWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::SegmentWriter.

25.249.4 Member Data Documentation

25.249.4.1 unsigned long gdcm::SurfaceWriter::NumberOfSurfaces [protected]

The documentation for this class was generated from the following file:

- gdcmSurfaceWriter.h

25.250 gdcm::SwapCode Class Reference

SwapCode representation.

```
#include <gdcmSwapCode.h>
```

Public Types

- enum SwapCodeType {
  Unknown = 0,
  LittleEndian = 1234,
  BigEndian = 4321,
  BadLittleEndian = 3412,
  BadBigEndian = 2143 }

Public Member Functions

- SwapCode (SwapCodeType sc=Unknown)
- operator SwapCode::SwapCodeType () const

Static Public Member Functions

- static const char * GetSwapCodeString (SwapCode const &sc)

Static Protected Member Functions

- static int GetIndex (SwapCode const &sc)

Friends

- std::ostream & operator<< (std::ostream &os, const SwapCode &sc)

25.250.1 Detailed Description

SwapCode representation.

Examples:

TestByteSwap.cxx.

25.250.2 Member Enumeration Documentation

25.250.2.1 enum gdcm::SwapCode::SwapCodeType

Enumerator:

Unknown

LittleEndian

BigEndian

BadLittleEndian
BadBigEndian

25.250.3 Constructor & Destructor Documentation

25.250.3.1 gdcm::SwapCode::SwapCode (SwapCodeType *sc* = Unknown) [inline]

25.250.4 Member Function Documentation

25.250.4.1 static int gdcm::SwapCode::GetIndex (SwapCode const & *sc*) [static], [protected]

25.250.4.2 static const char* gdcm::SwapCode::GetSwapCodeString (SwapCode const & *sc*) [static]

Referenced by gdcm::operator<<().

25.250.4.3 gdcm::SwapCode::operator SwapCode::SwapCodeType ( ) const [inline]

25.250.5 Friends And Related Function Documentation

25.250.5.1 std::ostream& operator<< (std::ostream & *os*, const SwapCode & *sc*) [friend]

The documentation for this class was generated from the following file:

- gdcmSwapCode.h

25.251 gdcm::SwapperDoOp Class Reference

```
#include <gdcmSwapper.h>
```

Static Public Member Functions

- template<typename T >
  static T Swap (T val)
- template<typename T >
  static void SwapArray (T *array, size_t n)

25.251.1 Member Function Documentation

25.251.1.1 template<typename T > static T gdcm::SwapperDoOp::Swap (T *val*) [static]

Referenced by gdcm::Item::Read().

25.251.1.2 template<typename T > static void gdcm::SwapperDoOp::SwapArray (T * *array*, size_t *n*) [inline], [static]

The documentation for this class was generated from the following file:

- gdcmSwapper.h

25.252 gdcm::SwapperNoOp Class Reference

```
#include <gdcmSwapper.h>
```

Static Public Member Functions

- template<typename T >
  static T Swap (T val)
- template<typename T >
  static void SwapArray (T *, size_t)

25.252.1 Detailed Description

Examples:

ReadExplicitLengthSQIVR.cxx.

25.252.2 Member Function Documentation

25.252.2.1 template<typename T > static T gdcm::SwapperNoOp::Swap (T *val*) [inline],[static]

Referenced by gdcm::EncodingImplementation< VR::VRBINARY >::Write().

25.252.2.2 template<typename T > static void gdcm::SwapperNoOp::SwapArray (T *, size_t) [inline],[static]

Referenced by gdcm::EncodingImplementation< VR::VRBINARY >::Read().

The documentation for this class was generated from the following file:

- gdcmSwapper.h

25.253 gdcm::System Class Reference

Class to do system operation.

```
#include <gdcmSystem.h>
```

Static Public Member Functions

- static bool DeleteDirectory (const char *source)

  remove a directory named source
- static size_t EncodeBytes (char *out, const unsigned char *data, int size)
- static bool FileExists (const char *filename)

  Check whether the specified file exist on the sytem.
- static bool FileIsDirectory (const char *name)

  Check whether the file specified is a directory:
- static bool FileIsSymlink (const char *name)

  Check whether name is a symlink.

- static size_t FileSize (const char ∗filename)
- static time_t FileTime (const char ∗filename)
- static bool FormatDateTime (char date[22], time_t t, long milliseconds=0)
- static bool GetCurrentDateTime (char date[22])
- static const char ∗ GetCurrentModuleFileName ()
- static const char ∗ GetCurrentProcessFileName ()
- static const char ∗ GetCurrentResourcesDirectory ()
- static const char ∗ GetCWD ()
- static bool GetHostName (char hostname[255])
- static const char ∗ GetLastSystemError ()

  Return the last error.
- static const char ∗ GetLocaleCharset ()

  return `locale charmap`
- static const char ∗ GetTimezoneOffsetFromUTC ()
- static bool MakeDirectory (const char ∗path)

  Create a directory name path.
- static bool ParseDateTime (time_t &timep, const char date[22])

  Parse a date stored as ASCII text into a time_t structured (discard millisecond if any)
- static bool ParseDateTime (time_t &timep, long &milliseconds, const char date[22])
- static bool RemoveFile (const char ∗source)

  remove a file named source
- static int StrCaseCmp (const char ∗s1, const char ∗s2)

  consistent func for C99 spec of strcasecmp/strncasecmp
- static int StrNCaseCmp (const char ∗s1, const char ∗s2, size_t n)
- static char ∗ StrTokR (char ∗ptr, const char ∗sep, char ∗∗end)

  strtok_r

Static Protected Member Functions

- static bool GetPermissions (const char ∗file, unsigned short &mode)

  NOT THREAD SAFE.
- static bool SetPermissions (const char ∗file, unsigned short mode)

25.253.1 Detailed Description

Class to do system operation.

OS independent functionalities

25.253.2 Member Function Documentation

25.253.2.1 static bool gdcm::System::DeleteDirectory (const char ∗ *source*) [static]

remove a directory named source

25.253.2.2 static size_t gdcm::System::EncodeBytes (char ∗ *out*, const unsigned char ∗ *data*, int *size*) [static]

Used internally by the UIDGenerator class to convert a uuid tape to a DICOM VR:UI type

25.253.2.3 static bool gdcm::System::FileExists (const char * *filename*) [static]

Check whether the specified file exist on the sytem.

Examples:

EncapsulateFileInRawData.cxx, gdcmorthoplanes.cxx, and MagnifyFile.cxx.

25.253.2.4 static bool gdcm::System::FileIsDirectory (const char * *name*) [static]

Check whether the file specified is a directory:

Examples:

gdcmorthoplanes.cxx, and threadgdcm.cxx.

25.253.2.5 static bool gdcm::System::FileIsSymlink (const char * *name*) [static]

Check whether name is a symlink.

25.253.2.6 static size_t gdcm::System::FileSize (const char * *filename*) [static]

Return the filesize. 0 if file does not exist.

Warning

you need to use FileExists to differentiate between empty file and missing file.
for very large size file and on system where size_t is not appropriate to store off_t value the function will return 0.

Examples:

CheckBigEndianBug.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, and EncapsulateFileInRawData.cxx.

25.253.2.7 static time_t gdcm::System::FileTime (const char * *filename*) [static]

Return the time of last modification of file 0 if the file does not exist

25.253.2.8 static bool gdcm::System::FormatDateTime (char *date[22]*, time_t *t*, long *milliseconds* = 0) [static]

format as ASCII text a time_t with milliseconds See VR::DT from DICOM PS 3.5 milliseconds is in the range [0, 999999]

25.253.2.9 static bool gdcm::System::GetCurrentDateTime (char *date[22]*) [static]

Return the current data time, and format it as ASCII text. This is simply a call to gettimeofday + FormatDateTime, since WIN32 do not have an implementation for gettimeofday, this is more portable. The call time(0) is not precise for our resolution

25.253.2.10 static const char* gdcm::System::GetCurrentModuleFileName ( ) [static]

Return the directory the current module is located: NOT THREAD SAFE

25.253.2.11 static const char* gdcm::System::GetCurrentProcessFileName ( ) [static]

Return the directory the current process (executable) is located: NOT THREAD SAFE

25.253.2.12 static const char* gdcm::System::GetCurrentResourcesDirectory ( ) [static]

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

25.253.2.13 static const char* gdcm::System::GetCWD ( ) [static]

Return current working directory Warning: if current working path is too long (>2048 bytes) the call will fail and call will return NULL NOT THREAD SAFE

25.253.2.14 static bool gdcm::System::GetHostName (char *hostname[255]*) [static]

Retrieve the hostname, only the first 255 byte are copyied. This may come handy to specify the Station Name

25.253.2.15 static const char* gdcm::System::GetLastSystemError ( ) [static]

Return the last error.

25.253.2.16 static const char* gdcm::System::GetLocaleCharset ( ) [static]

return `locale charmap`

25.253.2.17 static bool gdcm::System::GetPermissions (const char * *file*, unsigned short & *mode*) [static], [protected]

NOT THREAD SAFE.

25.253.2.18 static const char* gdcm::System::GetTimezoneOffsetFromUTC ( ) [static]

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe

25.253.2.19 static bool gdcm::System::MakeDirectory (const char * *path*) [static]

Create a directory name path.

25.253.2.20 static bool gdcm::System::ParseDateTime (time_t & *timep,* const char *date[22]*) [static]

Parse a date stored as ASCII text into a time_t structured (discard millisecond if any)

25.253.2.21 static bool gdcm::System::ParseDateTime (time_t & *timep,* long & *milliseconds,* const char *date[22]*) [static]

Parse a date stored as ASCII text into a time_t structured and millisecond

See also

FormatDateTime

25.253.2.22 static bool gdcm::System::RemoveFile (const char * *source*) [static]

remove a file named source

25.253.2.23 static bool gdcm::System::SetPermissions (const char * *file,* unsigned short *mode*) [static], [protected]

25.253.2.24 static int gdcm::System::StrCaseCmp (const char * *s1,* const char * *s2*) [static]

consistent func for C99 spec of strcasecmp/strncasecmp

Referenced by gdcm::PrivateTag::operator<().

25.253.2.25 static int gdcm::System::StrNCaseCmp (const char * *s1,* const char * *s2,* size_t *n*) [static]

Precondition

n != 0

25.253.2.26 static char* gdcm::System::StrTokR (char * *ptr,* const char * *sep,* char ** *end*) [static]

strtok_r

The documentation for this class was generated from the following file:

- gdcmSystem.h

25.254 gdcm::Table Class Reference

Table.

```
#include <gdcmTable.h>
```

Public Types

- typedef std::map< Tag, TableEntry > MapTableEntry

Public Member Functions

- Table ()
- ∼Table ()
- const TableEntry & GetTableEntry (const Tag &tag) const
- void InsertEntry (Tag const &tag, TableEntry const &te)

Friends

- std::ostream & operator<< (std::ostream &_os, const Table &_val)

25.254.1 Detailed Description

Table.

25.254.2 Member Typedef Documentation

25.254.2.1 typedef std::map<Tag, TableEntry> gdcm::Table::MapTableEntry

25.254.3 Constructor & Destructor Documentation

25.254.3.1 gdcm::Table::Table ( ) [inline]

25.254.3.2 gdcm::Table::∼Table ( ) [inline]

25.254.4 Member Function Documentation

25.254.4.1 const TableEntry& gdcm::Table::GetTableEntry (const Tag & *tag*) const [inline]

25.254.4.2 void gdcm::Table::InsertEntry (Tag const & *tag,* TableEntry const & *te*) [inline]

25.254.5 Friends And Related Function Documentation

25.254.5.1 std::ostream& operator<< (std::ostream & *_os,* const Table & *_val*) [friend]

The documentation for this class was generated from the following file:

- gdcmTable.h

25.255 gdcm::TableEntry Class Reference

TableEntry.

```
#include <gdcmTableEntry.h>
```

Public Member Functions

- TableEntry (const char ∗attribute=0, Type const &type=Type(), const char ∗des=0)
- ∼TableEntry ()

25.255.1 Detailed Description

TableEntry.

25.255.2 Constructor & Destructor Documentation

25.255.2.1 gdcm::TableEntry::TableEntry (const char * *attribute* = 0, Type const & *type* = Type (), const char * *des* = 0) [inline]

25.255.2.2 gdcm::TableEntry::~TableEntry ( ) [inline]

The documentation for this class was generated from the following file:

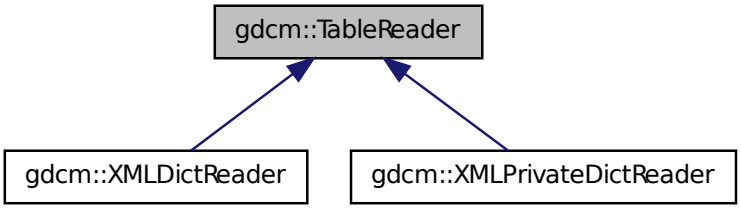
- gdcmTableEntry.h

25.256 gdcm::TableReader Class Reference

Class for representing a TableReader.

`#include <gdcmTableReader.h>`

Inheritance diagram for gdcm::TableReader:



Public Member Functions

- TableReader (Defs &defs)
- virtual ~TableReader ()
- virtual void CharacterDataHandler (const char *data, int length)
- virtual void EndElement (const char *name)
- const Defs & GetDefs () const
- const char * GetFilename ()
- void HandleIOD (const char **atts)
- void HandleIODEntry (const char **atts)
- void HandleMacro (const char **atts)
- void HandleMacroEntry (const char **atts)
- void HandleMacroEntryDescription (const char **atts)

- void HandleModule (const char **atts)
- void HandleModuleEntry (const char **atts)
- void HandleModuleEntryDescription (const char **atts)
- void HandleModuleInclude (const char **atts)
- int Read ()
- void SetFilename (const char *filename)
- virtual void StartElement (const char *name, const char **atts)

25.256.1 Detailed Description

Class for representing a TableReader.

Note

This class is an empty shell meant to be derived

25.256.2 Constructor & Destructor Documentation

25.256.2.1 gdcm::TableReader::TableReader (Defs & *defs*) [inline]

25.256.2.2 virtual gdcm::TableReader::~TableReader ( ) [inline], [virtual]

25.256.3 Member Function Documentation

25.256.3.1 virtual void gdcm::TableReader::CharacterDataHandler (const char * *data,* int *length*) [virtual]

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

25.256.3.2 virtual void gdcm::TableReader::EndElement (const char * *name*) [virtual]

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

25.256.3.3 const Defs& gdcm::TableReader::GetDefs ( ) const [inline]

25.256.3.4 const char* gdcm::TableReader::GetFilename ( ) [inline]

25.256.3.5 void gdcm::TableReader::HandleIOD (const char ** *atts*)

25.256.3.6 void gdcm::TableReader::HandleIODEntry (const char ** *atts*)

25.256.3.7 void gdcm::TableReader::HandleMacro (const char ** *atts*)

25.256.3.8 void gdcm::TableReader::HandleMacroEntry (const char ** *atts*)

25.256.3.9 void gdcm::TableReader::HandleMacroEntryDescription (const char ** *atts*)

25.256.3.10 void gdcm::TableReader::HandleModule (const char ** *atts*)

25.256.3.11 void gdcm::TableReader::HandleModuleEntry (const char ** *atts*)

25.256.3.12 void gdcm::TableReader::HandleModuleEntryDescription (const char ** *atts*)

25.256.3.13 void gdcm::TableReader::HandleModuleInclude (const char ** *atts*)

25.256.3.14 int gdcm::TableReader::Read ( )

25.256.3.15 void gdcm::TableReader::SetFilename (const char * *filename*) [inline]

25.256.3.16 virtual void gdcm::TableReader::StartElement (const char * *name,* const char ** *atts*) [virtual]

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

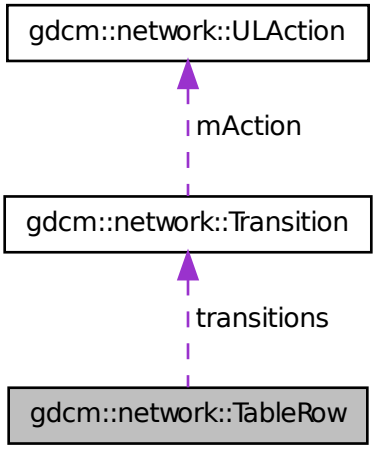
The documentation for this class was generated from the following file:

- gdcmTableReader.h

25.257 gdcm::network::TableRow Class Reference

`#include <gdcmULTransitionTable.h>`

Collaboration diagram for gdcm::network::TableRow:



Public Attributes

- Transition transitions [cMaxStateID]

25.257.1 Member Data Documentation

25.257.1.1 Transition gdcm::network::TableRow::transitions[cMaxStateID]

The documentation for this class was generated from the following file:

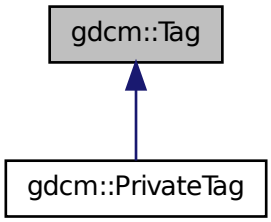
- gdcmULTransitionTable.h

25.258 gdcm::Tag Class Reference

Class to represent a DICOM Data Element (Attribute) Tag (Group, Element). Basically an uint32_t which can also be expressed as two uint16_t (group and element)

```
#include <gdcmTag.h>
```

Inheritance diagram for gdcm::Tag:



Public Member Functions

- Tag (uint16_t group, uint16_t element)

  Constructor with 2∗uint16_t.
- Tag (uint32_t tag=0)

  Constructor with 1∗uint32_t Prefer the cstor that takes two uint16_t.
- Tag (const Tag &_val)
- uint16_t GetElement () const

  Returns the 'Element number' of the given Tag.
- uint32_t GetElementTag () const

  Returns the full tag value of the given Tag.
- uint16_t GetGroup () const

  Returns the 'Group number' of the given Tag.
- uint32_t GetLength () const

  return the length of tag (read: size on disk)
- Tag GetPrivateCreator () const

  Return the Private Creator Data Element tag of a private data element.
- bool IsGroupLength () const

  return whether the tag correspond to a group length tag:
- bool IsGroupXX (const Tag &t) const

e.g 6002,3000 belong to groupXX: 6000,3000

- bool IsIllegal () const

  return if the tag is considered to be an illegal tag
- bool IsPrivate () const
- bool IsPrivateCreator () const
- bool IsPublic () const
- bool operator!= (const Tag &_val) const
- bool operator< (const Tag &_val) const
- bool operator<= (const Tag &t2) const
- Tag & operator= (const Tag &_val)
- bool operator== (const Tag &_val) const
- const uint16_t & operator[] (const unsigned int &_id) const

  Returns the Group or Element of the given Tag, depending on id (0/1)
- uint16_t & operator[] (const unsigned int &_id)

  Returns the Group or Element of the given Tag, depending on id (0/1)
- std::string PrintAsPipeSeparatedString () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)

  Read a tag from binary representation.
- bool ReadFromCommaSeparatedString (const char *str)
- bool ReadFromPipeSeparatedString (const char *str)
- void SetElement (uint16_t element)

  Sets the 'Element number' of the given Tag.
- void SetElementTag (uint16_t group, uint16_t element)

  Sets the 'Group number' & 'Element number' of the given Tag.
- void SetElementTag (uint32_t tag)

  Sets the full tag value of the given Tag.
- void SetGroup (uint16_t group)

  Sets the 'Group number' of the given Tag.
- void SetPrivateCreator (Tag const &t)

  Set private creator:
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

  Write a tag in binary rep.

Friends

- std::ostream & operator<< (std::ostream &_os, const Tag &_val)
- std::istream & operator>> (std::istream &_is, Tag &_val)

25.258.1 Detailed Description

Class to represent a DICOM Data Element (Attribute) Tag (Group, Element). Basically an uint32_t which can also be expressed as two uint16_t (group and element)

Note

DATA ELEMENT TAG: A unique identifier for a Data Element composed of an ordered pair of numbers (a Group Number followed by an Element Number). GROUP NUMBER: The first number in the ordered pair of numbers that makes up a Data Element Tag. ELEMENT NUMBER: The second number in the ordered pair of numbers that makes up a Data Element Tag.

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateJPIPDataSet.cxx, DumpToSQLITE3.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, PublicDict.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, rle2img.cxx, SimpleScanner.cxx, SortImage.cxx, StreamImageReaderTest.cxx, TraverseModules.cxx, and VolumeSorter.cxx.

25.258.2 Constructor & Destructor Documentation

25.258.2.1 gdcm::Tag::Tag (uint16_t *group*, uint16_t *element*) [inline]

Constructor with 2∗uint16_t.

25.258.2.2 gdcm::Tag::Tag (uint32_t *tag* = 0) [inline]

Constructor with 1∗uint32_t Prefer the cstor that takes two uint16_t.

25.258.2.3 gdcm::Tag::Tag (const Tag & *_val*) [inline]

References tag.

25.258.3 Member Function Documentation

25.258.3.1 uint16_t gdcm::Tag::GetElement ( ) const [inline]

Returns the 'Element number' of the given Tag.

Examples:

DuplicatePCDE.cxx, and PublicDict.cxx.

Referenced by gdcm::DataSet::ComputeGroupLength(), IsGroupXX(), gdcm::PrivateDict::PrintXML(), and SetPrivateCreator().

25.258.3.2 uint32_t gdcm::Tag::GetElementTag ( ) const [inline]

Returns the full tag value of the given Tag.

25.258.3.3 uint16_t gdcm::Tag::GetGroup ( ) const [inline]

Returns the 'Group number' of the given Tag.

Examples:

DuplicatePCDE.cxx, and GenAllVR.cxx.

Referenced by gdcm::DataSet::ComputeGroupLength(), gdcm::CommandDataSet::Insert(), gdcm::FileMetaInformation::Insert(), gdcm::DataSet::Insert(), IsGroupXX(), gdcm::PrivateDict::PrintXML(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

25.258.3.4 uint32_t gdcm::Tag::GetLength ( ) const [inline]

return the length of tag (read: size on disk)

25.258.3.5 Tag gdcm::Tag::GetPrivateCreator ( ) const [inline]

Return the Private Creator Data Element tag of a private data element.

References SetElement().

25.258.3.6 bool gdcm::Tag::IsGroupLength ( ) const [inline]

return whether the tag correspond to a group length tag:

25.258.3.7 bool gdcm::Tag::IsGroupXX (const Tag & *t*) const [inline]

e.g 6002,3000 belong to groupXX: 6000,3000

References GetElement(), GetGroup(), and IsPrivate().

25.258.3.8 bool gdcm::Tag::IsIllegal ( ) const [inline]

return if the tag is considered to be an illegal tag

25.258.3.9 bool gdcm::Tag::IsPrivate ( ) const [inline]

PRIVATE DATA ELEMENT: Additional Data Element, defined by an implementor, to communicate information that is not contained in Standard Data Elements. Private Data elements have odd Group Numbers.

Examples:

DuplicatePCDE.cxx.

Referenced by IsGroupXX(), and SetPrivateCreator().

25.258.3.10 bool gdcm::Tag::IsPrivateCreator ( ) const [inline]

Returns if tag is a Private Creator (xxxx,00yy), where xxxx is odd number and yy in [0x10,0xFF]

Examples:

DuplicatePCDE.cxx.

25.258.3.11 bool gdcm::Tag::IsPublic ( ) const [inline]

STANDARD DATA ELEMENT: A Data Element defined in the DICOM Standard, and therefore listed in the DICOM Data Element Dictionary in PS 3.6. Is the Tag from the Public dict...well the implementation is buggy it does not prove the element is indeed in the dict...

25.258.3.12 bool gdcm::Tag::operator!= (const Tag & *_val*) const [inline]

References tag.

25.258.3.13 bool gdcm::Tag::operator< (const Tag & *_val*) const [inline]

DICOM Standard expects the Data Element to be sorted by Tags All other comparison can be constructed from this one and operator ==

References tag, and tags.

25.258.3.14 bool gdcm::Tag::operator<= (const Tag & *t2*) const [inline]

25.258.3.15 Tag& gdcm::Tag::operator= (const Tag & *_val*) [inline]

References tag.

25.258.3.16 bool gdcm::Tag::operator== (const Tag & *_val*) const [inline]

References tag.

25.258.3.17 const uint16_t& gdcm::Tag::operator[] (const unsigned int & *_id*) const [inline]

Returns the Group or Element of the given Tag, depending on id (0/1)

25.258.3.18 uint16_t& gdcm::Tag::operator[] (const unsigned int & *_id*) [inline]

Returns the Group or Element of the given Tag, depending on id (0/1)

25.258.3.19 std::string gdcm::Tag::PrintAsPipeSeparatedString ( ) const

Print as a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

ReadFromPipeSeparatedString

25.258.3.20 template<typename TSwap > std::istream& gdcm::Tag::Read (std::istream & *is*) [inline]

Read a tag from binary representation.

25.258.3.21 bool gdcm::Tag::ReadFromCommaSeparatedString (const char * *str*)

Read from a comma separated string. This is a highly user oriented function, the string should be formated as: 1234,5678 to specify the tag (0x1234,0x5678) The notation comes from the DICOM standard, and is handy to use from a command line program

Reimplemented in gdcm::PrivateTag.

25.258.3.22 bool gdcm::Tag::ReadFromPipeSeparatedString (const char * *str*)

Read from a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

ReadFromCommaSeparatedString

25.258.3.23 void gdcm::Tag::SetElement (uint16_t *element*) [inline]

Sets the 'Element number' of the given Tag.

Examples:

DuplicatePCDE.cxx, and PublicDict.cxx.

Referenced by GetPrivateCreator(), and gdcm::operator>>().

25.258.3.24 void gdcm::Tag::SetElementTag (uint16_t *group,* uint16_t *element*) [inline]

Sets the 'Group number' & 'Element number' of the given Tag.

25.258.3.25 void gdcm::Tag::SetElementTag (uint32_t *tag*) [inline]

Sets the full tag value of the given Tag.

25.258.3.26 void gdcm::Tag::SetGroup (uint16_t *group*) [inline]

Sets the 'Group number' of the given Tag.

Referenced by gdcm::operator>>().

25.258.3.27 void gdcm::Tag::SetPrivateCreator (Tag const & *t*) [inline]

Set private creator:

Examples:

DuplicatePCDE.cxx.

References GetElement(), and IsPrivate().

25.258.3.28 template<typename TSwap > const std::ostream& gdcm::Tag::Write (std::ostream & *os*) const [inline]

Write a tag in binary rep.

Referenced by gdcm::SequenceOfFragments::Write(), gdcm::SequenceOfItems::Write(), and gdcm::Item::Write().

25.258.4 Friends And Related Function Documentation

25.258.4.1 std::ostream& operator<< (std::ostream & *_os,* const Tag & *_val*) [friend]

25.258.4.2 std::istream& operator>> (std::istream & *_is,* Tag & *_val*) [friend]

25.258.5 Member Data Documentation

25.258.5.1 char gdcm::Tag::bytes[4]

25.258.5.2 uint32_t gdcm::Tag::tag

Referenced by operator!=(), operator<(), operator=(), operator==(), and Tag().

25.258.5.3 uint16_t gdcm::Tag::tags[2]

Referenced by operator<().

The documentation for this class was generated from the following file:

- gdcmTag.h

25.259 gdcm::TagPath Class Reference

class to handle a path of tag.

```
#include <gdcmTagPath.h>
```

Public Member Functions

- TagPath ()
- ~TagPath ()
- bool ConstructFromString (const char *path)
- bool ConstructFromTagList (Tag const *l, unsigned int n)

Construct from a list of tags.

- void Print (std::ostream &) const
- bool Push (Tag const &t)
- bool Push (unsigned int itemnum)

Static Public Member Functions

- static bool IsValid (const char *path)

  Return if path is valid or not.

25.259.1 Detailed Description

class to handle a path of tag.

Any Resemblance to Existing XPath is Purely Coincidental `ftp://medical.nema.org/medical/dicom/supps/sup118-_pc.pdf`

25.259.2 Constructor & Destructor Documentation

25.259.2.1 gdcm::TagPath::TagPath ( )

25.259.2.2 gdcm::TagPath::~TagPath ( )

25.259.3 Member Function Documentation

25.259.3.1 bool gdcm::TagPath::ConstructFromString (const char * *path*)

"/0018,0018/"... No space allowed, comma is use to separate tag group from tag element and slash is used to separate tag return false if invalid

25.259.3.2 bool gdcm::TagPath::ConstructFromTagList (Tag const * *l*, unsigned int *n*)

Construct from a list of tags.

25.259.3.3 static bool gdcm::TagPath::IsValid (const char * *path*) [static]

Return if path is valid or not.

25.259.3.4 void gdcm::TagPath::Print (std::ostream &) const

25.259.3.5 bool gdcm::TagPath::Push (Tag const & *t*)

25.259.3.6 bool gdcm::TagPath::Push (unsigned int *itemnum*)

The documentation for this class was generated from the following file:

- gdcmTagPath.h

25.260 gdcm::Testing Class Reference

class for testing

`#include <gdcmTesting.h>`

Public Types

- typedef const char ∗const (∗ MD5DataImagesType)[2]
- typedef const char ∗const (∗ MediaStorageDataFilesType)[2]

  return the table that map the media storage (as string) of a filename (gdcmData)

Public Member Functions

- Testing ()
- ∼Testing ()
- void Print (std::ostream &os=std::cout)

  Print.

Static Public Member Functions

- static bool ComputeFileMD5 (const char ∗filename, char digest_str[33])
- static bool ComputeMD5 (const char ∗buffer, unsigned long buf_len, char digest_str[33])
- static const char ∗ GetDataExtraRoot ()

  Return the GDCM DATA EXTRA ROOT.
- static const char ∗ GetDataRoot ()

  Return the GDCM DATA ROOT.
- static const char ∗ GetFileName (unsigned int file)
- static const char ∗const ∗ GetFileNames ()

  return the table of fullpath to gdcmData DICOM files:
- static int GetLossyFlagFromFile (const char ∗filepath)
- static const char ∗const ∗ GetMD5DataImage (unsigned int file)
- static MD5DataImagesType GetMD5DataImages ()
- static const char ∗ GetMD5FromBrokenFile (const char ∗filepath)
- static const char ∗ GetMD5FromFile (const char ∗filepath)
- static const char ∗const ∗ GetMediaStorageDataFile (unsigned int file)
- static MediaStorageDataFilesType GetMediaStorageDataFiles ()
- static const char ∗ GetMediaStorageFromFile (const char ∗filepath)
- static unsigned int GetNumberOfFileNames ()
- static unsigned int GetNumberOfMD5DataImages ()
- static unsigned int GetNumberOfMediaStorageDataFiles ()
- static const char ∗ GetPixelSpacingDataRoot ()

  Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)
- static std::streamoff GetSelectedTagsOffsetFromFile (const char ∗filepath)
- static const char ∗ GetSourceDirectory ()
- static std::streamoff GetStreamOffsetFromFile (const char ∗filepath)
- static const char ∗ GetTempDirectory (const char ∗subdir=0)
- static const wchar_t ∗ GetTempDirectoryW (const wchar_t ∗subdir=0)

NOT THREAD SAFE.

- static const char ∗ GetTempFilename (const char ∗filename, const char ∗subdir=0)

  NOT THREAD SAFE.

- static const wchar_t ∗ GetTempFilenameW (const wchar_t ∗filename, const wchar_t ∗subdir=0)

  NOT THREAD SAFE.

25.260.1 Detailed Description

class for testing

this class is used for the nightly regression system for GDCM It makes heavily use of md5 computation

See also

gdcm::MD5 class for md5 computation

25.260.2 Member Typedef Documentation

25.260.2.1 typedef const char∗ const(∗ gdcm::Testing::MD5DataImagesType)[2]

return the table that map the md5 (as in md5sum) of the Pixel Data associated to a filename

25.260.2.2 typedef const char∗ const(∗ gdcm::Testing::MediaStorageDataFilesType)[2]

return the table that map the media storage (as string) of a filename (gdcmData)

25.260.3 Constructor & Destructor Documentation

25.260.3.1 gdcm::Testing::Testing ( ) [inline]

25.260.3.2 gdcm::Testing::∼Testing ( ) [inline]

25.260.4 Member Function Documentation

25.260.4.1 static bool gdcm::Testing::ComputeFileMD5 (const char ∗ *filename,* char *digest_str[33]*) [static]

25.260.4.2 static bool gdcm::Testing::ComputeMD5 (const char ∗ *buffer,* unsigned long *buf_len,* char *digest_str[33]*) [static]

MD5 stuff digest_str needs to be at least : strlen = [2∗16+1]; string will be \0 padded. (md5 are 32 bytes long) Testing is not meant to be shipped with an installed GDCM release, always prefer the gdcm::MD5 API when doing md5 computation.

25.260.4.3 static const char∗ gdcm::Testing::GetDataExtraRoot ( ) [static]

Return the GDCM DATA EXTRA ROOT.

Examples:

DiscriminateVolume.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

25.260.4.4 static const char* gdcm::Testing::GetDataRoot ( ) [static]

Return the GDCM DATA ROOT.

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, and MagnifyFile.cxx.

25.260.4.5 static const char* gdcm::Testing::GetFileName (unsigned int *file*) [static]

25.260.4.6 static const char* const* gdcm::Testing::GetFileNames ( ) [static]

return the table of fullpath to gdcmData DICOM files:

Examples:

TestReader.cxx.

25.260.4.7 static int gdcm::Testing::GetLossyFlagFromFile (const char * *filepath*) [static]

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

25.260.4.8 static const char* const* gdcm::Testing::GetMD5DataImage (unsigned int *file*) [static]

25.260.4.9 static MD5DataImagesType gdcm::Testing::GetMD5DataImages ( ) [static]

25.260.4.10 static const char* gdcm::Testing::GetMD5FromBrokenFile (const char * *filepath*) [static]

Return what should have been the md5 of file 'filepath' This is based on current GDCM implementation to decipher a broken DICOM file.

25.260.4.11 static const char* gdcm::Testing::GetMD5FromFile (const char * *filepath*) [static]

25.260.4.12 static const char* const* gdcm::Testing::GetMediaStorageDataFile (unsigned int *file*) [static]

25.260.4.13 static MediaStorageDataFilesType gdcm::Testing::GetMediaStorageDataFiles ( ) [static]

25.260.4.14 static const char* gdcm::Testing::GetMediaStorageFromFile (const char * *filepath*) [static]

Examples:

TestReader.cxx.

25.260.4.15 static unsigned int gdcm::Testing::GetNumberOfFileNames ( ) [static]

25.260.4.16 static unsigned int gdcm::Testing::GetNumberOfMD5DataImages ( ) [static]

25.260.4.17 static unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles ( ) [static]

25.260.4.18 static const char* gdcm::Testing::GetPixelSpacingDataRoot ( ) [static]

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

25.260.4.19 static std::streamoff gdcm::Testing::GetSelectedTagsOffsetFromFile (const char * *filepath*) [static]

Return the offset just after Pixel Data Length (7fe0,0000) if found. Otherwise the offset of the very first pixel cell in Pixel Data -1 if not found

25.260.4.20 static const char* gdcm::Testing::GetSourceDirectory ( ) [static]

25.260.4.21 static std::streamoff gdcm::Testing::GetStreamOffsetFromFile (const char * *filepath*) [static]

Return the offset of the very first pixel cell in the PixelData -1 if not found

25.260.4.22 static const char* gdcm::Testing::GetTempDirectory (const char * *subdir* = 0) [static]

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

25.260.4.23 static const wchar_t* gdcm::Testing::GetTempDirectoryW (const wchar_t * *subdir* = 0) [static]

NOT THREAD SAFE.

25.260.4.24 static const char* gdcm::Testing::GetTempFilename (const char * *filename*, const char * *subdir* = 0) [static]

NOT THREAD SAFE.

25.260.4.25 static const wchar_t* gdcm::Testing::GetTempFilenameW (const wchar_t * *filename*, const wchar_t * *subdir* = 0) [static]

NOT THREAD SAFE.

25.260.4.26 void gdcm::Testing::Print (std::ostream & *os* = std::cout)

Print.

The documentation for this class was generated from the following file:

- gdcmTesting.h

25.261 gdcm::Trace Class Reference

Trace.

```
#include <gdcmTrace.h>
```

Public Member Functions

- Trace ()
- ∼Trace ()

Static Public Member Functions

- static void DebugOff ()
- static void DebugOn ()
- static void ErrorOff ()
- static void ErrorOn ()
- static bool GetDebugFlag ()
- static bool GetErrorFlag ()
- static std::ostream & GetStream ()
- static bool GetWarningFlag ()
- static void SetDebug (bool debug)
- static void SetError (bool debug)
- static void SetStream (std::ostream &os)

  Explicitely set the ostream for gdcm::Trace to report to.
- static void SetWarning (bool debug)
- static void WarningOff ()
- static void WarningOn ()

25.261.1 Detailed Description

Trace.

Debug / Warning and Error are encapsulated in this class by default the Trace class will redirect any debug/warning/error to std::cerr. Unless SetStream was specified with another (open) stream.

25.261.2 Constructor & Destructor Documentation

25.261.2.1 gdcm::Trace::Trace ( )

25.261.2.2 gdcm::Trace::∼Trace ( )

25.261.3 Member Function Documentation

25.261.3.1 static void gdcm::Trace::DebugOff ( ) [static]

Examples:

TestReader.cxx.

25.261.3.2 static void gdcm::Trace::DebugOn ( ) [static]

Examples:

Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.261.3.3 static void gdcm::Trace::ErrorOff ( ) [static]

25.261.3.4 static void gdcm::Trace::ErrorOn ( ) [static]

25.261.3.5 static bool gdcm::Trace::GetDebugFlag ( ) [static]

25.261.3.6 static bool gdcm::Trace::GetErrorFlag ( ) [static]

25.261.3.7 static std::ostream& gdcm::Trace::GetStream ( ) [static]

25.261.3.8 static bool gdcm::Trace::GetWarningFlag ( ) [static]

25.261.3.9 static void gdcm::Trace::SetDebug (bool *debug*) [static]

Examples:

DumpToSQLITE3.cxx.

25.261.3.10 static void gdcm::Trace::SetError (bool *debug*) [static]

25.261.3.11 static void gdcm::Trace::SetStream (std::ostream & *os*) [static]

Explicitely set the ostream for gdcm::Trace to report to.

25.261.3.12 static void gdcm::Trace::SetWarning (bool *debug*) [static]

Examples:

DumpToSQLITE3.cxx.

25.261.3.13 static void gdcm::Trace::WarningOff ( ) [static]

Examples:

TestReader.cxx.

25.261.3.14 static void gdcm::Trace::WarningOn ( ) [static]

Examples:

Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

The documentation for this class was generated from the following file:

- gdcmTrace.h

25.262 gdcm::TransferSyntax Class Reference

Class to manipulate Transfer Syntax.

```
#include <gdcmTransferSyntax.h>
```

Public Types

- enum NegociatedType {
  Unknown = 0,
  Explicit,
  Implicit }
- enum TSType {
  ImplicitVRLittleEndian = 0,
  ImplicitVRBigEndianPrivateGE,
  ExplicitVRLittleEndian,
  DeflatedExplicitVRLittleEndian,
  ExplicitVRBigEndian,
  JPEGBaselineProcess1,
  JPEGExtendedProcess2_4,
  JPEGExtendedProcess3_5,
  JPEGSpectralSelectionProcess6_8,
  JPEGFullProgressionProcess10_12,
  JPEGLosslessProcess14,
  JPEGLosslessProcess14_1,
  JPEGLSLossless,
  JPEGLSNearLossless,
  JPEG2000Lossless,
  JPEG2000,
  JPIPReferenced,
  RLELossless,
  MPEG2MainProfile,
  ImplicitVRBigEndianACRNEMA,
  CT_private_ELE,
  TS_END }

Public Member Functions

- TransferSyntax (TSType type=ImplicitVRLittleEndian)
- bool CanStoreLossy () const
- NegociatedType GetNegociatedType () const
- const char ∗ GetString () const
- SwapCode GetSwapCode () const
- bool IsEncapsulated () const
- bool IsEncoded () const
- bool IsExplicit () const
- bool IsImplicit () const
- bool IsLossless () const
- bool IsLossy () const
- bool IsValid () const
- operator TSType () const

Static Public Member Functions

- static const char ∗ GetTSString (TSType ts)
- static TSType GetTSType (const char ∗str)

Friends

- std::ostream & operator<< (std::ostream &os, const TransferSyntax &ts)

25.262.1 Detailed Description

Class to manipulate Transfer Syntax.

Note

TRANSFER SYNTAX (Standard and Private): A set of encoding rules that allow Application Entities to unambiguously negotiate the encoding techniques (e.g., Data Element structure, byte ordering, compression) they are able to support, thereby allowing these Application Entities to communicate.

Todo : The implementation is completely retarded -> see gdcm::UIDs for a replacement We need: IsSupported We need preprocess of raw/xml file We need GetFullName()

Need a notion of Private Syntax. As defined in PS 3.5. Section 9.2

See also

UIDs

Examples:

GetJPEGSamplePrecision.cxx, and LargeVRDSExplicit.cxx.

25.262.2 Member Enumeration Documentation

25.262.2.1 enum gdcm::TransferSyntax::NegociatedType

Enumerator:

Unknown

Explicit

Implicit

25.262.2.2 enum gdcm::TransferSyntax::TSType

Enumerator:

ImplicitVRLittleEndian

ImplicitVRBigEndianPrivateGE

ExplicitVRLittleEndian

DeflatedExplicitVRLittleEndian

ExplicitVRBigEndian

JPEGBaselineProcess1

JPEGExtendedProcess2_4

JPEGExtendedProcess3_5

JPEGSpectralSelectionProcess6_8

JPEGFullProgressionProcess10_12

JPEGLosslessProcess14

JPEGLosslessProcess14_1

JPEGLSLossless

JPEGLSNearLossless

JPEG2000Lossless

JPEG2000

JPIPReferenced

RLELossless

MPEG2MainProfile

ImplicitVRBigEndianACRNEMA

CT_private_ELE

TS_END

25.262.3 Constructor & Destructor Documentation

25.262.3.1 gdcm::TransferSyntax::TransferSyntax (TSType *type* = ImplicitVRLittleEndian) [inline]

25.262.4 Member Function Documentation

25.262.4.1 bool gdcm::TransferSyntax::CanStoreLossy ( ) const

return if TransFer Syntax Allow storing of Lossy Pixel Data

25.262.4.2 NegociatedType gdcm::TransferSyntax::GetNegociatedType ( ) const

25.262.4.3 const char* gdcm::TransferSyntax::GetString ( ) const [inline]

References GetTSString().

25.262.4.4 SwapCode gdcm::TransferSyntax::GetSwapCode ( ) const

Deprecated Return the SwapCode associated with the Transfer Syntax. Be careful with the special GE private syntax the DataSet is written in little endian but the Pixel Data is in Big Endian.

25.262.4.5 static const char* gdcm::TransferSyntax::GetTSString (TSType *ts*) [static]

Examples:

LargeVRDSExplicit.cxx.

Referenced by GetString(), and gdcm::operator<<().

25.262.4.6 static TSType gdcm::TransferSyntax::GetTSType (const char * *str*) [static]

25.262.4.7 bool gdcm::TransferSyntax::IsEncapsulated ( ) const

Examples:

ExtractIconFromFile.cxx.

25.262.4.8 bool gdcm::TransferSyntax::IsEncoded ( ) const

25.262.4.9 bool gdcm::TransferSyntax::IsExplicit ( ) const

25.262.4.10 bool gdcm::TransferSyntax::IsImplicit ( ) const

25.262.4.11 bool gdcm::TransferSyntax::IsLossless ( ) const

Return if the transfer syntax algorithm is a lossless algorithm

25.262.4.12 bool gdcm::TransferSyntax::IsLossy ( ) const

Return if the transfer syntax algorithm is a lossy algorithm

25.262.4.13 bool gdcm::TransferSyntax::IsValid ( ) const [inline]

25.262.4.14 gdcm::TransferSyntax::operator TSType ( ) const [inline]

25.262.5 Friends And Related Function Documentation

25.262.5.1 std::ostream& operator<< (std::ostream & *os*, const TransferSyntax & *ts*) [friend]

The documentation for this class was generated from the following file:

- gdcmTransferSyntax.h

25.263 gdcm::network::TransferSyntaxSub Class Reference

TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

```
#include <gdcmTransferSyntaxSub.h>
```

Public Member Functions

- TransferSyntaxSub ()
- const char * GetName () const
- bool operator== (const TransferSyntaxSub &ts) const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetName (const char *name)

- void SetNameFromUID (UIDs::TSName tsname)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.263.1 Detailed Description

TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

TODO what is the goal of :

Table 9-19 TRANSFER SYNTAX SUB-ITEM FIELDS

25.263.2 Constructor & Destructor Documentation

25.263.2.1 gdcm::network::TransferSyntaxSub::TransferSyntaxSub ( )

25.263.3 Member Function Documentation

25.263.3.1 const char* gdcm::network::TransferSyntaxSub::GetName ( ) const [inline]

25.263.3.2 bool gdcm::network::TransferSyntaxSub::operator== (const TransferSyntaxSub & *ts*) const [inline]

25.263.3.3 void gdcm::network::TransferSyntaxSub::Print (std::ostream & *os*) const

25.263.3.4 std::istream& gdcm::network::TransferSyntaxSub::Read (std::istream & *is*)

25.263.3.5 void gdcm::network::TransferSyntaxSub::SetName (const char * *name*)

25.263.3.6 void gdcm::network::TransferSyntaxSub::SetNameFromUID (UIDs::TSName *tsname*)

25.263.3.7 size_t gdcm::network::TransferSyntaxSub::Size ( ) const

25.263.3.8 const std::ostream& gdcm::network::TransferSyntaxSub::Write (std::ostream & *os*) const

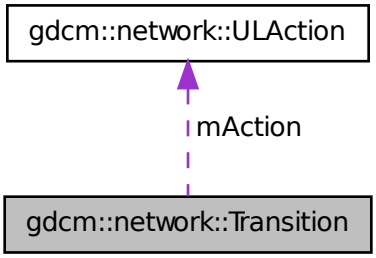
The documentation for this class was generated from the following file:

- gdcmTransferSyntaxSub.h

25.264 gdcm::network::Transition Struct Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcm::network::Transition:



Public Member Functions

- Transition ()
- Transition (int inEndState, ULAction ∗inAction)
- ∼Transition ()

Static Public Member Functions

- static Transition ∗ MakeNew (int inEndState, ULAction ∗inAction)

Public Attributes

- ULAction ∗ mAction
- int mEnd

25.264.1 Constructor & Destructor Documentation

25.264.1.1 gdcm::network::Transition::Transition ( ) [inline]

References gdcm::network::eStaDoesNotExist, mAction, and mEnd.

Referenced by MakeNew().

25.264.1.2 gdcm::network::Transition::∼Transition ( ) [inline]

References mAction.

25.264.1.3 gdcm::network::Transition::Transition (int *inEndState*, ULAction ∗ *inAction*) [inline]

References mAction, and mEnd.

25.264.2 Member Function Documentation

25.264.2.1 static Transition* gdcm::network::Transition::MakeNew (int *inEndState,* ULAction * *inAction*) [inline], [static]

References Transition().

25.264.3 Member Data Documentation

25.264.3.1 ULAction* gdcm::network::Transition::mAction

Referenced by Transition(), and ~Transition().

25.264.3.2 int gdcm::network::Transition::mEnd

Referenced by Transition().

The documentation for this struct was generated from the following file:

- gdcmULTransitionTable.h

25.265 gdcm::Type Class Reference

Type.

```
#include <gdcmType.h>
```

Public Types

- enum TypeType {
  T1 = 0,
  T1C,
  T2,
  T2C,
  T3,
  UNKNOWN }

Public Member Functions

- Type (TypeType type=UNKNOWN)
- operator TypeType () const

Static Public Member Functions

- static const char * GetTypeString (TypeType type)
- static TypeType GetTypeType (const char *type)

Friends

- std::ostream & operator<< (std::ostream &os, const Type &vr)

25.265.1 Detailed Description

Type.

Note

PS 3.5 7.4 DATA ELEMENT TYPE 7.4.1 TYPE 1 REQUIRED DATA ELEMENTS 7.4.2 TYPE 1C CONDITIONAL DATA ELEMENTS 7.4.3 TYPE 2 REQUIRED DATA ELEMENTS 7.4.4 TYPE 2C CONDITIONAL DATA ELEMENTS 7.4.5 TYPE 3 OPTIONAL DATA ELEMENTS

The intent of Type 2 Data Elements is to allow a zero length to be conveyed when the operator or application does not know its value or has a specific reason for not specifying its value. It is the intent that the device should support these Data Elements.

Examples:

TraverseModules.cxx.

25.265.2 Member Enumeration Documentation

25.265.2.1 enum gdcm::Type::TypeType

Enumerator:

T1

T1C

T2

T2C

T3

UNKNOWN

25.265.3 Constructor & Destructor Documentation

25.265.3.1 gdcm::Type::Type (TypeType *type* = UNKNOWN) [inline]

25.265.4 Member Function Documentation

25.265.4.1 static const char* gdcm::Type::GetTypeString (TypeType *type*) [static]

Referenced by gdcm::operator<<().

25.265.4.2 static TypeType gdcm::Type::GetTypeType (const char * *type*) [static]

Referenced by gdcm::ModuleEntry::ModuleEntry().

25.265.4.3 gdcm::Type::operator TypeType ( ) const [inline]

25.265.5 Friends And Related Function Documentation

25.265.5.1 std::ostream& operator<< (std::ostream & *os*, const Type & *vr*) [friend]

The documentation for this class was generated from the following file:

- gdcmType.h

25.266 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

Public Attributes

- char Internal [64+1]

Friends

- std::ostream & operator<< (std::ostream &_os, const UI &_val)

25.266.1 Friends And Related Function Documentation

25.266.1.1 std::ostream& operator<< (std::ostream & *_os*, const UI & *_val*) [friend]

25.266.2 Member Data Documentation

25.266.2.1 char gdcm::UI::Internal[64+1]

Referenced by gdcm::operator<<().

The documentation for this struct was generated from the following file:

- gdcmVR.h

25.267 gdcm::UIDGenerator Class Reference

Class for generating unique UID.

```
#include <gdcmUIDGenerator.h>
```

Public Member Functions

- UIDGenerator ()

  By default the root of a UID is a GDCM Root...
- const char * Generate ()

Static Public Member Functions

- static const char ∗ GetGDCMUID ()

  Return the default (GDCM) root UID:
- static const char ∗ GetRoot ()
- static bool IsValid (const char ∗uid)
- static void SetRoot (const char ∗root)

Static Protected Member Functions

- static bool GenerateUUID (unsigned char ∗uuid_data)

25.267.1 Detailed Description

Class for generating unique UID.

Note

bla Usage: When constructing a Series or Study UID, user *has* to keep around the UID, otherwise the UID Generator will simply forget the value and create a new UID.

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_-Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GetSubSequenceData.-cxx, StreamImageReaderTest.cxx, and uid_unique.cxx.

25.267.2 Constructor & Destructor Documentation

25.267.2.1 gdcm::UIDGenerator::UIDGenerator ( ) [inline]

By default the root of a UID is a GDCM Root...

25.267.3 Member Function Documentation

25.267.3.1 const char∗ gdcm::UIDGenerator::Generate ( )

Internally uses a std::string, so two calls have the same pointer ! save into a std::string In summary do not write code like that: const char ∗uid1 = uid.Generate(); const char ∗uid2 = uid.Generate(); since uid1 == uid2

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_-Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, StreamImageReader-Test.cxx, and uid_unique.cxx.

25.267.3.2 static bool gdcm::UIDGenerator::GenerateUUID (unsigned char ∗ *uuid_data*) [static],[protected]

25.267.3.3 static const char∗ gdcm::UIDGenerator::GetGDCMUID ( ) [static]

Return the default (GDCM) root UID:

25.267.3.4 static const char* gdcm::UIDGenerator::GetRoot ( ) [static]

25.267.3.5 static bool gdcm::UIDGenerator::IsValid (const char * *uid*) [static]

Find out if the string is a valid UID or not

Todo : Move that in DataStructureAndEncoding (see FileMetaInformation::CheckFileMetaInformation)

25.267.3.6 static void gdcm::UIDGenerator::SetRoot (const char * *root*) [static]

The current implementation in GDCM make use of the UUID implementation (RFC 4122) and has been successfully been tested for a root of size 26 bytes. Any longer root should work (the ::Generate() function will return a string), but will truncate the high bits of the 128bits UUID until the generated string fits on 64 bits. The authors disclaims any responsabitlity for garanteeing uniqueness of UIDs when the root is longer than 26 bytes.

Examples:

uid_unique.cxx.

The documentation for this class was generated from the following file:

- gdcmUIDGenerator.h

25.268 gdcm::UIDs Class Reference

all known uids

`#include <gdcmUIDs.h>`

Public Types

- typedef const char *const (* TransferSyntaxStringsType)[2]
- enum TSName {

VerificationSOPClass = 1,
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM = 2,
ExplicitVRLittleEndian = 3,
DeflatedExplicitVRLittleEndian = 4,
ExplicitVRBigEndian = 5,
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression = 6,
JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only = 7,
JPEGExtendedProcess35Retired = 8,
JPEGSpectralSelectionNonHierarchicalProcess68Retired = 9,
JPEGSpectralSelectionNonHierarchicalProcess79Retired = 10,
JPEGFullProgressionNonHierarchicalProcess1012Retired = 11,
JPEGFullProgressionNonHierarchicalProcess1113Retired = 12,
JPEGLosslessNonHierarchicalProcess14 = 13,
JPEGLosslessNonHierarchicalProcess15Retired = 14,
JPEGExtendedHierarchicalProcess1618Retired = 15,
JPEGExtendedHierarchicalProcess1719Retired = 16,
JPEGSpectralSelectionHierarchicalProcess2022Retired = 17,
JPEGSpectralSelectionHierarchicalProcess2123Retired = 18,
JPEGFullProgressionHierarchicalProcess2426Retired = 19,
JPEGFullProgressionHierarchicalProcess2527Retired = 20,
JPEGLosslessHierarchicalProcess28Retired = 21,
JPEGLosslessHierarchicalProcess29Retired = 22,
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLossless-

JPEGImageCompression = 23,
JPEGLSLosslessImageCompression = 24,
JPEGLSLossyNearLosslessImageCompression = 25,
JPEG2000ImageCompressionLosslessOnly = 26,
JPEG2000ImageCompression = 27,
JPEG2000Part2MulticomponentImageCompressionLosslessOnly = 28,
JPEG2000Part2MulticomponentImageCompression = 29,
JPIPReferenced = 30,
JPIPReferencedDeflate = 31,
MPEG2MainProfileMainLevel = 32,
RLELossless = 33,
RFC2557MIMEencapsulation = 34,
XMLEncoding = 35,
MediaStorageDirectoryStorage = 36,
TalairachBrainAtlasFrameofReference = 37,
SPM2T1FrameofReference = 38,
SPM2T2FrameofReference = 39,
SPM2PDFrameofReference = 40,
SPM2EPIFrameofReference = 41,
SPM2FILT1FrameofReference = 42,
SPM2PETFrameofReference = 43,
SPM2TRANSMFrameofReference = 44,
SPM2SPECTFrameofReference = 45,
SPM2GRAYFrameofReference = 46,
SPM2WHITEFrameofReference = 47,
SPM2CSFFrameofReference = 48,
SPM2BRAINMASKFrameofReference = 49,
SPM2AVG305T1FrameofReference = 50,
SPM2AVG152T1FrameofReference = 51,
SPM2AVG152T2FrameofReference = 52,
SPM2AVG152PDFrameofReference = 53,
SPM2SINGLESUBJT1FrameofReference = 54,
ICBM452T1FrameofReference = 55,
ICBMSingleSubjectMRIFrameofReference = 56,
BasicStudyContentNotificationSOPClassRetired = 57,
StorageCommitmentPushModelSOPClass = 58,
StorageCommitmentPushModelSOPInstance = 59,
StorageCommitmentPullModelSOPClassRetired = 60,
StorageCommitmentPullModelSOPInstanceRetired = 61,
ProceduralEventLoggingSOPClass = 62,
ProceduralEventLoggingSOPInstance = 63,
SubstanceAdministrationLoggingSOPClass = 64,
SubstanceAdministrationLoggingSOPInstance = 65,
DICOMUIDRegistry = 66,
DICOMControlledTerminology = 67,
DICOMApplicationContextName = 68,
DetachedPatientManagementSOPClassRetired = 69,
DetachedPatientManagementMetaSOPClassRetired = 70,
DetachedVisitManagementSOPClassRetired = 71,
DetachedStudyManagementSOPClassRetired = 72,
StudyComponentManagementSOPClassRetired = 73,
ModalityPerformedProcedureStepSOPClass = 74,
ModalityPerformedProcedureStepRetrieveSOPClass = 75,
ModalityPerformedProcedureStepNotificationSOPClass = 76,
DetachedResultsManagementSOPClassRetired = 77,
DetachedResultsManagementMetaSOPClassRetired = 78,
DetachedStudyManagementMetaSOPClassRetired = 79,
DetachedInterpretationManagementSOPClassRetired = 80,
StorageServiceClass = 81,
BasicFilmSessionSOPClass = 82,

VLWholeSlideMicroscopyImageStorage }

- enum TSType {

uid_1_2_840_10008_1_1 = 1,
uid_1_2_840_10008_1_2 = 2,
uid_1_2_840_10008_1_2_1 = 3,
uid_1_2_840_10008_1_2_1_99 = 4,
uid_1_2_840_10008_1_2_2 = 5,
uid_1_2_840_10008_1_2_4_50 = 6,
uid_1_2_840_10008_1_2_4_51 = 7,
uid_1_2_840_10008_1_2_4_52 = 8,
uid_1_2_840_10008_1_2_4_53 = 9,
uid_1_2_840_10008_1_2_4_54 = 10,
uid_1_2_840_10008_1_2_4_55 = 11,
uid_1_2_840_10008_1_2_4_56 = 12,
uid_1_2_840_10008_1_2_4_57 = 13,
uid_1_2_840_10008_1_2_4_58 = 14,
uid_1_2_840_10008_1_2_4_59 = 15,
uid_1_2_840_10008_1_2_4_60 = 16,
uid_1_2_840_10008_1_2_4_61 = 17,
uid_1_2_840_10008_1_2_4_62 = 18,
uid_1_2_840_10008_1_2_4_63 = 19,
uid_1_2_840_10008_1_2_4_64 = 20,
uid_1_2_840_10008_1_2_4_65 = 21,
uid_1_2_840_10008_1_2_4_66 = 22,
uid_1_2_840_10008_1_2_4_70 = 23,
uid_1_2_840_10008_1_2_4_80 = 24,
uid_1_2_840_10008_1_2_4_81 = 25,
uid_1_2_840_10008_1_2_4_90 = 26,
uid_1_2_840_10008_1_2_4_91 = 27,
uid_1_2_840_10008_1_2_4_92 = 28,
uid_1_2_840_10008_1_2_4_93 = 29,
uid_1_2_840_10008_1_2_4_94 = 30,
uid_1_2_840_10008_1_2_4_95 = 31,
uid_1_2_840_10008_1_2_4_100 = 32,
uid_1_2_840_10008_1_2_5 = 33,
uid_1_2_840_10008_1_2_6_1 = 34,
uid_1_2_840_10008_1_2_6_2 = 35,
uid_1_2_840_10008_1_3_10 = 36,
uid_1_2_840_10008_1_4_1_1 = 37,
uid_1_2_840_10008_1_4_1_2 = 38,
uid_1_2_840_10008_1_4_1_3 = 39,
uid_1_2_840_10008_1_4_1_4 = 40,
uid_1_2_840_10008_1_4_1_5 = 41,
uid_1_2_840_10008_1_4_1_6 = 42,
uid_1_2_840_10008_1_4_1_7 = 43,
uid_1_2_840_10008_1_4_1_8 = 44,
uid_1_2_840_10008_1_4_1_9 = 45,
uid_1_2_840_10008_1_4_1_10 = 46,
uid_1_2_840_10008_1_4_1_11 = 47,
uid_1_2_840_10008_1_4_1_12 = 48,
uid_1_2_840_10008_1_4_1_13 = 49,
uid_1_2_840_10008_1_4_1_14 = 50,
uid_1_2_840_10008_1_4_1_15 = 51,
uid_1_2_840_10008_1_4_1_16 = 52,
uid_1_2_840_10008_1_4_1_17 = 53,
uid_1_2_840_10008_1_4_1_18 = 54,
uid_1_2_840_10008_1_4_2_1 = 55,
uid_1_2_840_10008_1_4_2_2 = 56,
uid_1_2_840_10008_1_9 = 57,
uid_1_2_840_10008_1_20_1 = 58,
uid_1_2_840_10008_1_20_1_1 = 59,
uid_1_2_840_10008_1_20_2 = 60,

uid_1_2_840_10008_5_1_4_1_1_77_1_6 }

Public Member Functions

- const char ∗ GetName () const
- const char ∗ GetString () const
- operator TSType () const
- bool SetFromUID (const char ∗str)

Static Public Member Functions

- static unsigned int GetNumberOfTransferSyntaxStrings ()
- static const char ∗const ∗ GetTransferSyntaxString (unsigned int ts)
- static TransferSyntaxStringsType GetTransferSyntaxStrings ()
- static const char ∗ GetUIDName (unsigned int ts)
- static const char ∗ GetUIDString (unsigned int ts)

25.268.1 Detailed Description

all known uids

Examples:

GenerateStandardSOPClasses.cxx.

25.268.2 Member Typedef Documentation

25.268.2.1 typedef const char∗ const(∗ gdcm::UIDs::TransferSyntaxStringsType)[2]

25.268.3 Member Enumeration Documentation

25.268.3.1 enum gdcm::UIDs::TSName

Enumerator:

VerificationSOPClass

ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM

ExplicitVRLittleEndian

DeflatedExplicitVRLittleEndian

ExplicitVRBigEndian

JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression

JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only

JPEGExtendedProcess35Retired

JPEGSpectralSelectionNonHierarchicalProcess68Retired

JPEGSpectralSelectionNonHierarchicalProcess79Retired

JPEGFullProgressionNonHierarchicalProcess1012Retired

JPEGFullProgressionNonHierarchicalProcess1113Retired

JPEGLosslessNonHierarchicalProcess14

JPEGLosslessNonHierarchicalProcess15Retired

JPEGExtendedHierarchicalProcess1618Retired

JPEGExtendedHierarchicalProcess1719Retired

JPEGSpectralSelectionHierarchicalProcess2022Retired

JPEGSpectralSelectionHierarchicalProcess2123Retired

JPEGFullProgressionHierarchicalProcess2426Retired

JPEGFullProgressionHierarchicalProcess2527Retired

JPEGLosslessHierarchicalProcess28Retired

JPEGLosslessHierarchicalProcess29Retired

JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImage

JPEGLSLosslessImageCompression

JPEGLSLossyNearLosslessImageCompression

JPEG2000ImageCompressionLosslessOnly

JPEG2000ImageCompression

JPEG2000Part2MulticomponentImageCompressionLosslessOnly

JPEG2000Part2MulticomponentImageCompression

JPIPReferenced

JPIPReferencedDeflate

MPEG2MainProfileMainLevel

RLELossless

RFC2557MIMEencapsulation

XMLEncoding

MediaStorageDirectoryStorage

TalairachBrainAtlasFrameofReference

SPM2T1FrameofReference

SPM2T2FrameofReference

SPM2PDFrameofReference

SPM2EPIFrameofReference

SPM2FILT1FrameofReference

SPM2PETFrameofReference

SPM2TRANSMFrameofReference

SPM2SPECTFrameofReference

SPM2GRAYFrameofReference

SPM2WHITEFrameofReference

SPM2CSFFrameofReference

SPM2BRAINMASKFrameofReference

SPM2AVG305T1FrameofReference

SPM2AVG152T1FrameofReference

SPM2AVG152T2FrameofReference

SPM2AVG152PDFrameofReference

SPM2SINGLESUBJT1FrameofReference

ICBM452T1FrameofReference

ICBMSingleSubjectMRIFrameofReference

BasicStudyContentNotificationSOPClassRetired

StorageCommitmentPushModelSOPClass

StorageCommitmentPushModelSOPInstance

StorageCommitmentPullModelSOPClassRetired

StorageCommitmentPullModelSOPInstanceRetired

ProceduralEventLoggingSOPClass

ProceduralEventLoggingSOPInstance

SubstanceAdministrationLoggingSOPClass

SubstanceAdministrationLoggingSOPInstance

DICOMUIDRegistry

DICOMControlledTerminology

DICOMApplicationContextName

DetachedPatientManagementSOPClassRetired

DetachedPatientManagementMetaSOPClassRetired

DetachedVisitManagementSOPClassRetired

DetachedStudyManagementSOPClassRetired

StudyComponentManagementSOPClassRetired

ModalityPerformedProcedureStepSOPClass

ModalityPerformedProcedureStepRetrieveSOPClass

ModalityPerformedProcedureStepNotificationSOPClass

DetachedResultsManagementSOPClassRetired

DetachedResultsManagementMetaSOPClassRetired

DetachedStudyManagementMetaSOPClassRetired

DetachedInterpretationManagementSOPClassRetired

StorageServiceClass

BasicFilmSessionSOPClass

BasicFilmBoxSOPClass

BasicGrayscaleImageBoxSOPClass

BasicColorImageBoxSOPClass

ReferencedImageBoxSOPClassRetired

BasicGrayscalePrintManagementMetaSOPClass

ReferencedGrayscalePrintManagementMetaSOPClassRetired

PrintJobSOPClass

BasicAnnotationBoxSOPClass

PrinterSOPClass

PrinterConfigurationRetrievalSOPClass

PrinterSOPInstance

PrinterConfigurationRetrievalSOPInstance

BasicColorPrintManagementMetaSOPClass

ReferencedColorPrintManagementMetaSOPClassRetired

VOILUTBoxSOPClass

PresentationLUTSOPClass

ImageOverlayBoxSOPClassRetired

BasicPrintImageOverlayBoxSOPClassRetired

PrintQueueSOPInstanceRetired

PrintQueueManagementSOPClassRetired

StoredPrintStorageSOPClassRetired

HardcopyGrayscaleImageStorageSOPClassRetired

HardcopyColorImageStorageSOPClassRetired

PullPrintRequestSOPClassRetired

PullStoredPrintManagementMetaSOPClassRetired

MediaCreationManagementSOPClassUID

ComputedRadiographyImageStorage

DigitalXRayImageStorageForPresentation

DigitalXRayImageStorageForProcessing

DigitalMammographyXRayImageStorageForPresentation

DigitalMammographyXRayImageStorageForProcessing

DigitalIntraoralXRayImageStorageForPresentation

DigitalIntraoralXRayImageStorageForProcessing

CTImageStorage

EnhancedCTImageStorage

UltrasoundMultiframeImageStorageRetired

UltrasoundMultiframeImageStorage

MRImageStorage

EnhancedMRImageStorage

MRSpectroscopyStorage

NuclearMedicineImageStorageRetired

UltrasoundImageStorageRetired

UltrasoundImageStorage

SecondaryCaptureImageStorage

MultiframeSingleBitSecondaryCaptureImageStorage

MultiframeGrayscaleByteSecondaryCaptureImageStorage

MultiframeGrayscaleWordSecondaryCaptureImageStorage

MultiframeTrueColorSecondaryCaptureImageStorage

StandaloneOverlayStorageRetired

StandaloneCurveStorageRetired

WaveformStorageTrialRetired

GeneralECGWaveformStorage

AmbulatoryECGWaveformStorage

HemodynamicWaveformStorage

CardiacElectrophysiologyWaveformStorage

BasicVoiceAudioWaveformStorage

StandaloneModalityLUTStorageRetired

StandaloneVOILUTStorageRetired

GrayscaleSoftcopyPresentationStateStorageSOPClass

ColorSoftcopyPresentationStateStorageSOPClass

PseudoColorSoftcopyPresentationStateStorageSOPClass

BlendingSoftcopyPresentationStateStorageSOPClass

XRayAngiographicImageStorage

EnhancedXAImageStorage

XRayRadiofluoroscopicImageStorage

EnhancedXRFImageStorage

XRay3DAngiographicImageStorage

XRay3DCraniofacialImageStorage

XRayAngiographicBiPlaneImageStorageRetired

NuclearMedicineImageStorage

RawDataStorage

SpatialRegistrationStorage

SpatialFiducialsStorage

DeformableSpatialRegistrationStorage

SegmentationStorage

RealWorldValueMappingStorage

VLImageStorageTrialRetired

VLMultiframeImageStorageTrialRetired

VLEndoscopicImageStorage

VideoEndoscopicImageStorage

VLMicroscopicImageStorage

VideoMicroscopicImageStorage

VLSlideCoordinatesMicroscopicImageStorage

VLPhotographicImageStorage

VideoPhotographicImageStorage

OphthalmicPhotography8BitImageStorage

OphthalmicPhotography16BitImageStorage

StereometricRelationshipStorage

OphthalmicTomographyImageStorage

TextSRStorageTrialRetired

AudioSRStorageTrialRetired

DetailSRStorageTrialRetired

ComprehensiveSRStorageTrialRetired

BasicTextSRStorage

EnhancedSRStorage

ComprehensiveSRStorage

ProcedureLogStorage

MammographyCADSRStorage

KeyObjectSelectionDocumentStorage

ChestCADSRStorage

XRayRadiationDoseSRStorage

EncapsulatedPDFStorage

EncapsulatedCDAStorage

PositronEmissionTomographyImageStorage

StandalonePETCurveStorageRetired

RTImageStorage

RTDoseStorage

RTStructureSetStorage

RTBeamsTreatmentRecordStorage

RTPlanStorage

RTBrachyTreatmentRecordStorage

RTTreatmentSummaryRecordStorage

RTIonPlanStorage

RTIonBeamsTreatmentRecordStorage

PatientRootQueryRetrieveInformationModelFIND

PatientRootQueryRetrieveInformationModelMOVE

PatientRootQueryRetrieveInformationModelGET

StudyRootQueryRetrieveInformationModelFIND

StudyRootQueryRetrieveInformationModelMOVE

StudyRootQueryRetrieveInformationModelGET

PatientStudyOnlyQueryRetrieveInformationModelFINDRetired

PatientStudyOnlyQueryRetrieveInformationModelMOVERetired

PatientStudyOnlyQueryRetrieveInformationModelGETRetired

ModalityWorklistInformationModelFIND

GeneralPurposeWorklistInformationModelFIND

GeneralPurposeScheduledProcedureStepSOPClass

GeneralPurposePerformedProcedureStepSOPClass

GeneralPurposeWorklistManagementMetaSOPClass

InstanceAvailabilityNotificationSOPClass

RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft

RTConventionalMachineVerificationSupplement74FrozenDraft

RTIonMachineVerificationSupplement74FrozenDraft

UnifiedWorklistandProcedureStepServiceClass

UnifiedProcedureStepPushSOPClass

UnifiedProcedureStepWatchSOPClass

UnifiedProcedureStepPullSOPClass

UnifiedProcedureStepEventSOPClass

UnifiedWorklistandProcedureStepSOPInstance

GeneralRelevantPatientInformationQuery

BreastImagingRelevantPatientInformationQuery

CardiacRelevantPatientInformationQuery

HangingProtocolStorage

HangingProtocolInformationModelFIND

HangingProtocolInformationModelMOVE

ProductCharacteristicsQuerySOPClass

SubstanceApprovalQuerySOPClass

dicomDeviceName

dicomDescription

dicomManufacturer

dicomManufacturerModelName

dicomSoftwareVersion

dicomVendorData

dicomAETitle

dicomNetworkConnectionReference

dicomApplicationCluster

dicomAssociationInitiator

dicomAssociationAcceptor

dicomHostname

dicomPort

dicomSOPClass

dicomTransferRole

dicomTransferSyntax

dicomPrimaryDeviceType

dicomRelatedDeviceReference

dicomPreferredCalledAETitle

dicomTLSCyphersuite

dicomAuthorizedNodeCertificateReference

dicomThisNodeCertificateReference

dicomInstalled

dicomStationName

dicomDeviceSerialNumber

dicomInstitutionName

dicomInstitutionAddress

dicomInstitutionDepartmentName

dicomIssuerOfPatientID

dicomPreferredCallingAETitle

dicomSupportedCharacterSet

dicomConfigurationRoot

dicomDevicesRoot

dicomUniqueAETitlesRegistryRoot

dicomDevice

dicomNetworkAE

dicomNetworkConnection

dicomUniqueAETitle

dicomTransferCapability

VLWholeSlideMicroscopyImageStorage

25.268.3.2 enum gdcm::UIDs::TSType

Enumerator:

uid_1_2_840_10008_1_1

uid_1_2_840_10008_1_2

uid_1_2_840_10008_1_2_1

uid_1_2_840_10008_1_2_1_99

uid_1_2_840_10008_1_2_2

uid_1_2_840_10008_1_2_4_50

uid_1_2_840_10008_1_2_4_51

uid_1_2_840_10008_1_2_4_52

uid_1_2_840_10008_1_2_4_53

uid_1_2_840_10008_1_2_4_54

uid_1_2_840_10008_1_2_4_55

uid_1_2_840_10008_1_2_4_56

uid_1_2_840_10008_1_2_4_57

uid_1_2_840_10008_1_2_4_58

uid_1_2_840_10008_1_2_4_59

uid_1_2_840_10008_1_2_4_60

uid_1_2_840_10008_1_2_4_61

uid_1_2_840_10008_1_2_4_62

uid_1_2_840_10008_1_2_4_63

uid_1_2_840_10008_1_2_4_64

uid_1_2_840_10008_1_2_4_65

uid_1_2_840_10008_1_2_4_66

uid_1_2_840_10008_1_2_4_70

uid_1_2_840_10008_1_2_4_80

uid_1_2_840_10008_1_2_4_81

uid_1_2_840_10008_1_2_4_90

uid_1_2_840_10008_1_2_4_91

uid_1_2_840_10008_1_2_4_92

uid_1_2_840_10008_1_2_4_93

uid_1_2_840_10008_1_2_4_94

uid_1_2_840_10008_1_2_4_95

uid_1_2_840_10008_1_2_4_100

uid_1_2_840_10008_1_2_5
uid_1_2_840_10008_1_2_6_1
uid_1_2_840_10008_1_2_6_2
uid_1_2_840_10008_1_3_10
uid_1_2_840_10008_1_4_1_1
uid_1_2_840_10008_1_4_1_2
uid_1_2_840_10008_1_4_1_3
uid_1_2_840_10008_1_4_1_4
uid_1_2_840_10008_1_4_1_5
uid_1_2_840_10008_1_4_1_6
uid_1_2_840_10008_1_4_1_7
uid_1_2_840_10008_1_4_1_8
uid_1_2_840_10008_1_4_1_9
uid_1_2_840_10008_1_4_1_10
uid_1_2_840_10008_1_4_1_11
uid_1_2_840_10008_1_4_1_12
uid_1_2_840_10008_1_4_1_13
uid_1_2_840_10008_1_4_1_14
uid_1_2_840_10008_1_4_1_15
uid_1_2_840_10008_1_4_1_16
uid_1_2_840_10008_1_4_1_17
uid_1_2_840_10008_1_4_1_18
uid_1_2_840_10008_1_4_2_1
uid_1_2_840_10008_1_4_2_2
uid_1_2_840_10008_1_9
uid_1_2_840_10008_1_20_1
uid_1_2_840_10008_1_20_1_1
uid_1_2_840_10008_1_20_2
uid_1_2_840_10008_1_20_2_1
uid_1_2_840_10008_1_40
uid_1_2_840_10008_1_40_1
uid_1_2_840_10008_1_42
uid_1_2_840_10008_1_42_1
uid_1_2_840_10008_2_6_1
uid_1_2_840_10008_2_16_4
uid_1_2_840_10008_3_1_1_1
uid_1_2_840_10008_3_1_2_1_1
uid_1_2_840_10008_3_1_2_1_4
uid_1_2_840_10008_3_1_2_2_1
uid_1_2_840_10008_3_1_2_3_1
uid_1_2_840_10008_3_1_2_3_2
uid_1_2_840_10008_3_1_2_3_3

uid_1_2_840_10008_3_1_2_3_4
uid_1_2_840_10008_3_1_2_3_5
uid_1_2_840_10008_3_1_2_5_1
uid_1_2_840_10008_3_1_2_5_4
uid_1_2_840_10008_3_1_2_5_5
uid_1_2_840_10008_3_1_2_6_1
uid_1_2_840_10008_4_2
uid_1_2_840_10008_5_1_1_1
uid_1_2_840_10008_5_1_1_2
uid_1_2_840_10008_5_1_1_4
uid_1_2_840_10008_5_1_1_4_1
uid_1_2_840_10008_5_1_1_4_2
uid_1_2_840_10008_5_1_1_9
uid_1_2_840_10008_5_1_1_9_1
uid_1_2_840_10008_5_1_1_14
uid_1_2_840_10008_5_1_1_15
uid_1_2_840_10008_5_1_1_16
uid_1_2_840_10008_5_1_1_16_376
uid_1_2_840_10008_5_1_1_17
uid_1_2_840_10008_5_1_1_17_376
uid_1_2_840_10008_5_1_1_18
uid_1_2_840_10008_5_1_1_18_1
uid_1_2_840_10008_5_1_1_22
uid_1_2_840_10008_5_1_1_23
uid_1_2_840_10008_5_1_1_24
uid_1_2_840_10008_5_1_1_24_1
uid_1_2_840_10008_5_1_1_25
uid_1_2_840_10008_5_1_1_26
uid_1_2_840_10008_5_1_1_27
uid_1_2_840_10008_5_1_1_29
uid_1_2_840_10008_5_1_1_30
uid_1_2_840_10008_5_1_1_31
uid_1_2_840_10008_5_1_1_32
uid_1_2_840_10008_5_1_1_33
uid_1_2_840_10008_5_1_4_1_1_1
uid_1_2_840_10008_5_1_4_1_1_1_1
uid_1_2_840_10008_5_1_4_1_1_1_1_1
uid_1_2_840_10008_5_1_4_1_1_1_2
uid_1_2_840_10008_5_1_4_1_1_1_2_1
uid_1_2_840_10008_5_1_4_1_1_1_3
uid_1_2_840_10008_5_1_4_1_1_1_3_1
uid_1_2_840_10008_5_1_4_1_1_2

uid_1_2_840_10008_5_1_4_1_1_2_1
uid_1_2_840_10008_5_1_4_1_1_3
uid_1_2_840_10008_5_1_4_1_1_3_1
uid_1_2_840_10008_5_1_4_1_1_4
uid_1_2_840_10008_5_1_4_1_1_4_1
uid_1_2_840_10008_5_1_4_1_1_4_2
uid_1_2_840_10008_5_1_4_1_1_5
uid_1_2_840_10008_5_1_4_1_1_6
uid_1_2_840_10008_5_1_4_1_1_6_1
uid_1_2_840_10008_5_1_4_1_1_7
uid_1_2_840_10008_5_1_4_1_1_7_1
uid_1_2_840_10008_5_1_4_1_1_7_2
uid_1_2_840_10008_5_1_4_1_1_7_3
uid_1_2_840_10008_5_1_4_1_1_7_4
uid_1_2_840_10008_5_1_4_1_1_8
uid_1_2_840_10008_5_1_4_1_1_9
uid_1_2_840_10008_5_1_4_1_1_9_1
uid_1_2_840_10008_5_1_4_1_1_9_1_1
uid_1_2_840_10008_5_1_4_1_1_9_1_2
uid_1_2_840_10008_5_1_4_1_1_9_1_3
uid_1_2_840_10008_5_1_4_1_1_9_2_1
uid_1_2_840_10008_5_1_4_1_1_9_3_1
uid_1_2_840_10008_5_1_4_1_1_9_4_1
uid_1_2_840_10008_5_1_4_1_1_10
uid_1_2_840_10008_5_1_4_1_1_11
uid_1_2_840_10008_5_1_4_1_1_11_1
uid_1_2_840_10008_5_1_4_1_1_11_2
uid_1_2_840_10008_5_1_4_1_1_11_3
uid_1_2_840_10008_5_1_4_1_1_11_4
uid_1_2_840_10008_5_1_4_1_1_12_1
uid_1_2_840_10008_5_1_4_1_1_12_1_1
uid_1_2_840_10008_5_1_4_1_1_12_2
uid_1_2_840_10008_5_1_4_1_1_12_2_1
uid_1_2_840_10008_5_1_4_1_1_13_1_1
uid_1_2_840_10008_5_1_4_1_1_13_1_2
uid_1_2_840_10008_5_1_4_1_1_12_3
uid_1_2_840_10008_5_1_4_1_1_20
uid_1_2_840_10008_5_1_4_1_1_66
uid_1_2_840_10008_5_1_4_1_1_66_1
uid_1_2_840_10008_5_1_4_1_1_66_2
uid_1_2_840_10008_5_1_4_1_1_66_3
uid_1_2_840_10008_5_1_4_1_1_66_4

uid_1_2_840_10008_5_1_4_1_1_67
uid_1_2_840_10008_5_1_4_1_1_77_1
uid_1_2_840_10008_5_1_4_1_1_77_2
uid_1_2_840_10008_5_1_4_1_1_77_1_1
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1
uid_1_2_840_10008_5_1_4_1_1_77_1_2
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1
uid_1_2_840_10008_5_1_4_1_1_77_1_3
uid_1_2_840_10008_5_1_4_1_1_77_1_4
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4
uid_1_2_840_10008_5_1_4_1_1_88_1
uid_1_2_840_10008_5_1_4_1_1_88_2
uid_1_2_840_10008_5_1_4_1_1_88_3
uid_1_2_840_10008_5_1_4_1_1_88_4
uid_1_2_840_10008_5_1_4_1_1_88_11
uid_1_2_840_10008_5_1_4_1_1_88_22
uid_1_2_840_10008_5_1_4_1_1_88_33
uid_1_2_840_10008_5_1_4_1_1_88_40
uid_1_2_840_10008_5_1_4_1_1_88_50
uid_1_2_840_10008_5_1_4_1_1_88_59
uid_1_2_840_10008_5_1_4_1_1_88_65
uid_1_2_840_10008_5_1_4_1_1_88_67
uid_1_2_840_10008_5_1_4_1_1_104_1
uid_1_2_840_10008_5_1_4_1_1_104_2
uid_1_2_840_10008_5_1_4_1_1_128
uid_1_2_840_10008_5_1_4_1_1_129
uid_1_2_840_10008_5_1_4_1_1_481_1
uid_1_2_840_10008_5_1_4_1_1_481_2
uid_1_2_840_10008_5_1_4_1_1_481_3
uid_1_2_840_10008_5_1_4_1_1_481_4
uid_1_2_840_10008_5_1_4_1_1_481_5
uid_1_2_840_10008_5_1_4_1_1_481_6
uid_1_2_840_10008_5_1_4_1_1_481_7
uid_1_2_840_10008_5_1_4_1_1_481_8
uid_1_2_840_10008_5_1_4_1_1_481_9
uid_1_2_840_10008_5_1_4_1_2_1_1
uid_1_2_840_10008_5_1_4_1_2_1_2
uid_1_2_840_10008_5_1_4_1_2_1_3

uid_1_2_840_10008_5_1_4_1_2_2_1
uid_1_2_840_10008_5_1_4_1_2_2_2
uid_1_2_840_10008_5_1_4_1_2_2_3
uid_1_2_840_10008_5_1_4_1_2_3_1
uid_1_2_840_10008_5_1_4_1_2_3_2
uid_1_2_840_10008_5_1_4_1_2_3_3
uid_1_2_840_10008_5_1_4_31
uid_1_2_840_10008_5_1_4_32_1
uid_1_2_840_10008_5_1_4_32_2
uid_1_2_840_10008_5_1_4_32_3
uid_1_2_840_10008_5_1_4_32
uid_1_2_840_10008_5_1_4_33
uid_1_2_840_10008_5_1_4_34_1
uid_1_2_840_10008_5_1_4_34_2
uid_1_2_840_10008_5_1_4_34_3
uid_1_2_840_10008_5_1_4_34_4
uid_1_2_840_10008_5_1_4_34_4_1
uid_1_2_840_10008_5_1_4_34_4_2
uid_1_2_840_10008_5_1_4_34_4_3
uid_1_2_840_10008_5_1_4_34_4_4
uid_1_2_840_10008_5_1_4_34_5
uid_1_2_840_10008_5_1_4_37_1
uid_1_2_840_10008_5_1_4_37_2
uid_1_2_840_10008_5_1_4_37_3
uid_1_2_840_10008_5_1_4_38_1
uid_1_2_840_10008_5_1_4_38_2
uid_1_2_840_10008_5_1_4_38_3
uid_1_2_840_10008_5_1_4_41
uid_1_2_840_10008_5_1_4_42
uid_1_2_840_10008_15_0_3_1
uid_1_2_840_10008_15_0_3_2
uid_1_2_840_10008_15_0_3_3
uid_1_2_840_10008_15_0_3_4
uid_1_2_840_10008_15_0_3_5
uid_1_2_840_10008_15_0_3_6
uid_1_2_840_10008_15_0_3_7
uid_1_2_840_10008_15_0_3_8
uid_1_2_840_10008_15_0_3_9
uid_1_2_840_10008_15_0_3_10
uid_1_2_840_10008_15_0_3_11
uid_1_2_840_10008_15_0_3_12
uid_1_2_840_10008_15_0_3_13

uid_1_2_840_10008_15_0_3_14

uid_1_2_840_10008_15_0_3_15

uid_1_2_840_10008_15_0_3_16

uid_1_2_840_10008_15_0_3_17

uid_1_2_840_10008_15_0_3_18

uid_1_2_840_10008_15_0_3_19

uid_1_2_840_10008_15_0_3_20

uid_1_2_840_10008_15_0_3_21

uid_1_2_840_10008_15_0_3_22

uid_1_2_840_10008_15_0_3_23

uid_1_2_840_10008_15_0_3_24

uid_1_2_840_10008_15_0_3_25

uid_1_2_840_10008_15_0_3_26

uid_1_2_840_10008_15_0_3_27

uid_1_2_840_10008_15_0_3_28

uid_1_2_840_10008_15_0_3_29

uid_1_2_840_10008_15_0_3_30

uid_1_2_840_10008_15_0_3_31

uid_1_2_840_10008_15_0_4_1

uid_1_2_840_10008_15_0_4_2

uid_1_2_840_10008_15_0_4_3

uid_1_2_840_10008_15_0_4_4

uid_1_2_840_10008_15_0_4_5

uid_1_2_840_10008_15_0_4_6

uid_1_2_840_10008_15_0_4_7

uid_1_2_840_10008_15_0_4_8

uid_1_2_840_10008_5_1_4_1_1_77_1_6

25.268.4 Member Function Documentation

25.268.4.1 const char* gdcm::UIDs::GetName ( ) const

When object is Initialize function return the well known name associated with uid return NULL when not initialized

Examples:

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

25.268.4.2 static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ( ) [static]

25.268.4.3 const char* gdcm::UIDs::GetString ( ) const

When object is Initialize function return the uid return NULL when not initialized

Examples:

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

25.268.4.4 static const char* const* gdcm::UIDs::GetTransferSyntaxString (unsigned int *ts*) [static]

25.268.4.5 static TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings ( ) [static]

25.268.4.6 static const char* gdcm::UIDs::GetUIDName (unsigned int *ts*) [static]

25.268.4.7 static const char* gdcm::UIDs::GetUIDString (unsigned int *ts*) [static]

25.268.4.8 gdcm::UIDs::operator TSType ( ) const [inline]

25.268.4.9 bool gdcm::UIDs::SetFromUID (const char * *str*)

Initialize object from a string (a uid number) return false on error, and internal state is set to 0

Examples:

GenerateStandardSOPClasses.cxx.

The documentation for this class was generated from the following file:

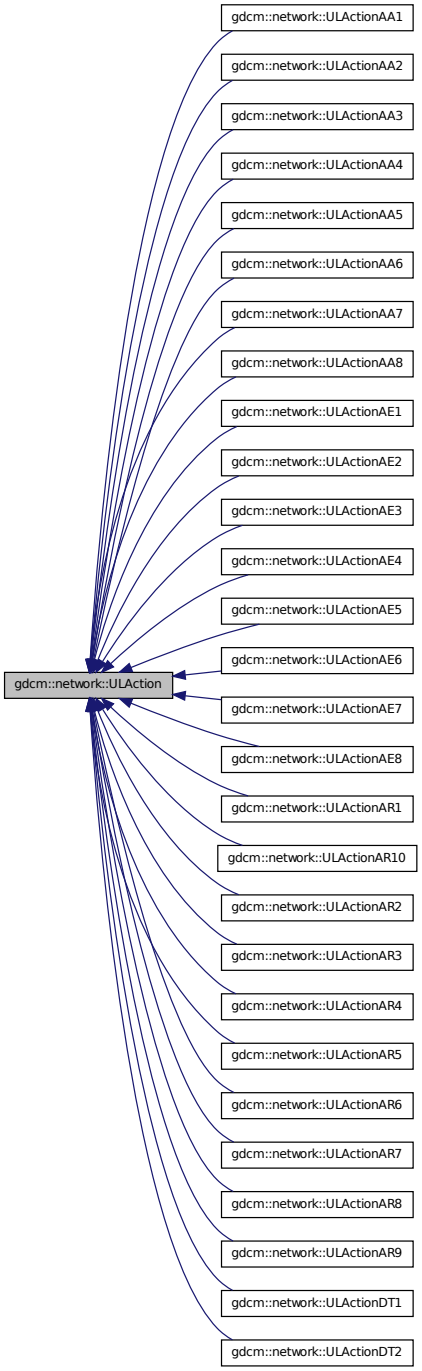
- gdcmUIDs.h

25.269 gdcm::network::ULAction Class Reference

ULAction A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.

```
#include <gdcmULAction.h>
```

Inheritance diagram for gdcm::network::ULAction:



Public Member Functions

- ULAction ()
- virtual ~ULAction ()

- virtual EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)=0

25.269.1 Detailed Description

ULAction A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.

Essentially, the ULConnectionManager will take this object, determined from the current ULState of the ULConnection, and pass the ULConnection object to the ULAction. The ULAction will then invoke whatever necessary commands are required by a given action.

The result of a ULAction is a ULEvent (ie, what happened as a result of the action).

This ULEvent is passed to the ULState, so that the transition to the next state can occur.

Actions are associated with Payloads– be thos filestreams, AETitles to establish connections, whatever. The actual parameters that the user will pass via an action will come through a Payload object, which should, in itself, be some gdcm-based object (but not all objects can be payloads; sending a single dataelement as a payload isn't meaningful). As such, each action has its own particular payload.

For the sake of keeping files together, both the particular payload class and the action class will be defined in the same header file. Payloads should JUST be data (or streams), NO METHODS.

Some actions perform changes that should raise events on the local system, and some actions perform changes that will require waiting for events from the remote system.

Therefore, this base action has been modified so that those events are set by each action. When the event loop runs an action, it will then test to see if a local event was raised by the action, and if so, perform the appropriate subsequent action. If the action requires waiting for a response from the remote system, then the event loop will sit there (presumably with the ARTIM timer running) and wait for a response from the remote system. Once a response is obtained, then the the rest of the state transitions can happen.

25.269.2 Constructor & Destructor Documentation

25.269.2.1 gdcm::network::ULAction::ULAction ( ) [inline]

25.269.2.2 virtual gdcm::network::ULAction::~ULAction ( ) [inline], [virtual]

25.269.3 Member Function Documentation

25.269.3.1 virtual EStateID gdcm::network::ULAction::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [pure virtual]

Implemented in gdcm::network::ULActionAR10, gdcm::network::ULActionAR9, gdcm::network::ULActionAE8, gdcm::network::ULActionAA8, gdcm::network::ULActionAR8, gdcm::network::ULActionAE7, gdcm::network::ULActionAA7, gdcm::network::ULActionAR7, gdcm::network::ULActionAE6, gdcm::network::ULActionAA6, gdcm::network::ULActionAR6, gdcm::network::ULActionAA5, gdcm::network::ULActionAE5, gdcm::network::ULActionAR5, gdcm::network::ULActionAA4, gdcm::network::ULActionAE4, gdcm::network::ULActionAR4, gdcm::network::ULActionAA3, gdcm::network::ULActionAE3, gdcm::network::ULActionAR3, gdcm::network::ULActionAA2, gdcm::network::ULActionAE2, gdcm::network::ULActionAR2, gdcm::network::ULActionDT2, gdcm::network::ULActionAA1, gdcm::network::ULActionAE1, gdcm::network::ULActionAR1, and gdcm::network::ULActionDT1.

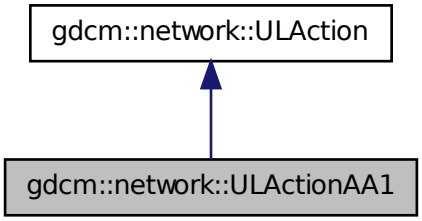
The documentation for this class was generated from the following file:

- gdcmULAction.h

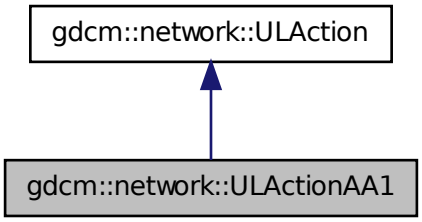
25.270 gdcm::network::ULActionAA1 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA1:



Collaboration diagram for gdcm::network::ULActionAA1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.270.1 Member Function Documentation

25.270.1.1 EStateID gdcm::network::ULActionAA1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

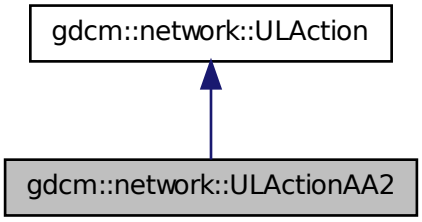
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

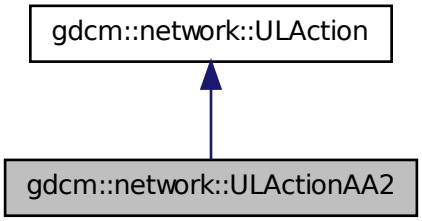
25.271 gdcm::network::ULActionAA2 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA2:



Collaboration diagram for gdcm::network::ULActionAA2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor-Event, EEventID &outRaisedEvent)

25.271.1 Member Function Documentation

25.271.1.1 EStateID gdcm::network::ULActionAA2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

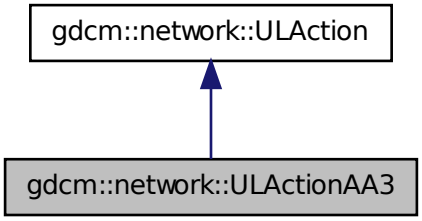
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

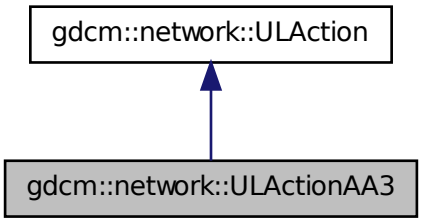
25.272 gdcm::network::ULActionAA3 Class Reference

`#include <gdcmULActionAA.h>`

Inheritance diagram for gdcm::network::ULActionAA3:



Collaboration diagram for gdcm::network::ULActionAA3:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.272.1 Member Function Documentation

25.272.1.1 EStateID gdcm::network::ULActionAA3::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

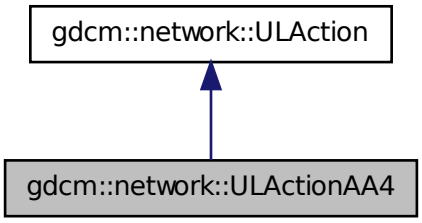
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

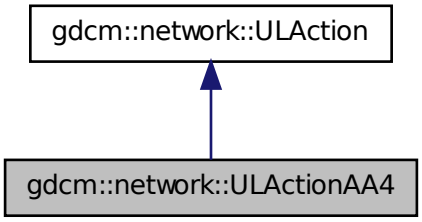
25.273 gdcm::network::ULActionAA4 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA4:



Collaboration diagram for gdcm::network::ULActionAA4:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.273.1 Member Function Documentation

25.273.1.1 EStateID gdcm::network::ULActionAA4::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

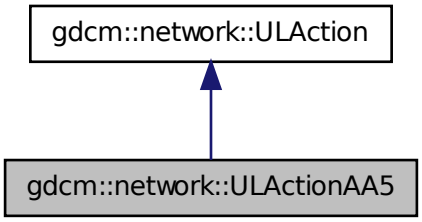
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

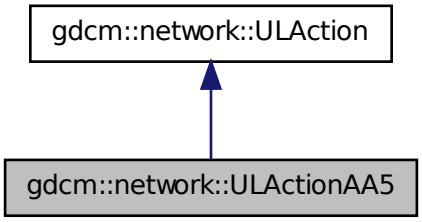
25.274 gdcm::network::ULActionAA5 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA5:



Collaboration diagram for gdcm::network::ULActionAA5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.274.1 Member Function Documentation

25.274.1.1 EStateID gdcm::network::ULActionAA5::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

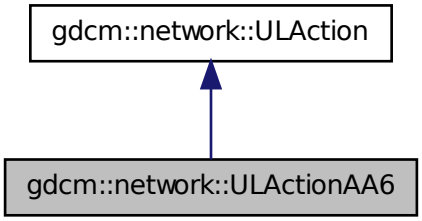
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

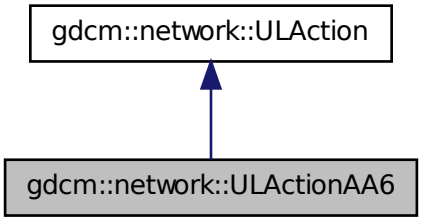
25.275 gdcm::network::ULActionAA6 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA6:



Collaboration diagram for gdcm::network::ULActionAA6:



Public Member Functions

- EStateID PerformAction (Subject ∗s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor-Event, EEventID &outRaisedEvent)

25.275.1 Member Function Documentation

25.275.1.1 EStateID gdcm::network::ULActionAA6::PerformAction (Subject ∗ *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

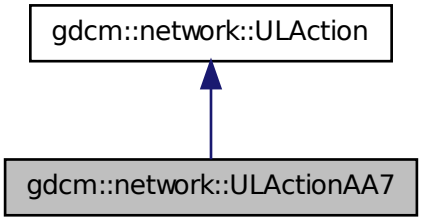
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

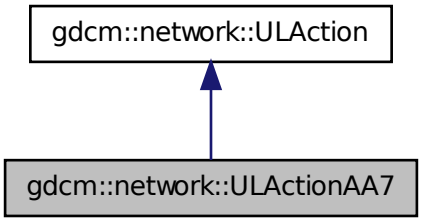
25.276 gdcm::network::ULActionAA7 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA7:



Collaboration diagram for gdcm::network::ULActionAA7:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.276.1 Member Function Documentation

25.276.1.1 EStateID gdcm::network::ULActionAA7::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

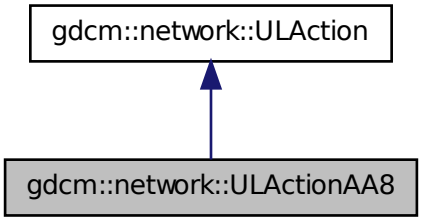
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

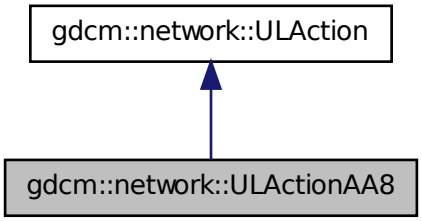
25.277 gdcm::network::ULActionAA8 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA8:



Collaboration diagram for gdcm::network::ULActionAA8:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.277.1 Member Function Documentation

25.277.1.1 EStateID gdcm::network::ULActionAA8::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

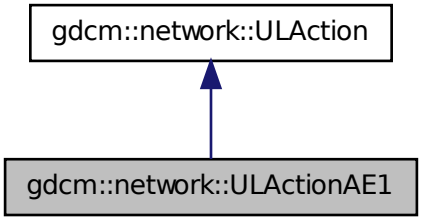
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

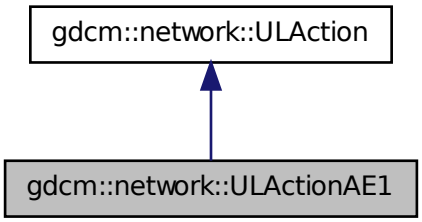
25.278 gdcm::network::ULActionAE1 Class Reference

`#include <gdcmULActionAE.h>`

Inheritance diagram for gdcm::network::ULActionAE1:



Collaboration diagram for gdcm::network::ULActionAE1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.278.1 Member Function Documentation

25.278.1.1 EStateID gdcm::network::ULActionAE1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

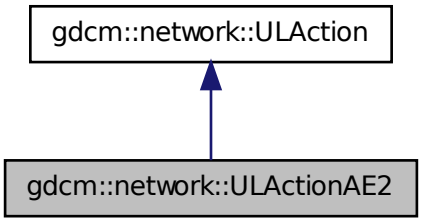
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

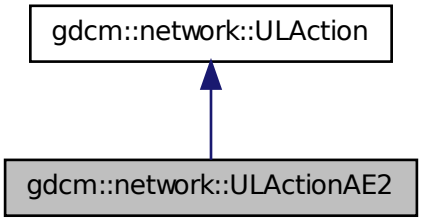
25.279 gdcm::network::ULActionAE2 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE2:



Collaboration diagram for gdcm::network::ULActionAE2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.279.1 Member Function Documentation

25.279.1.1 EStateID gdcm::network::ULActionAE2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

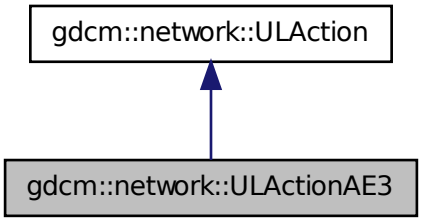
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

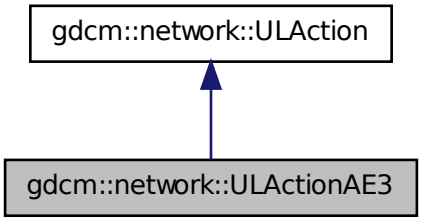
25.280 gdcm::network::ULActionAE3 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE3:



Collaboration diagram for gdcm::network::ULActionAE3:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.280.1 Member Function Documentation

25.280.1.1 EStateID gdcm::network::ULActionAE3::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

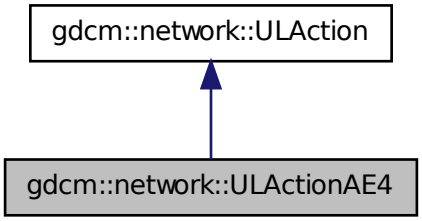
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

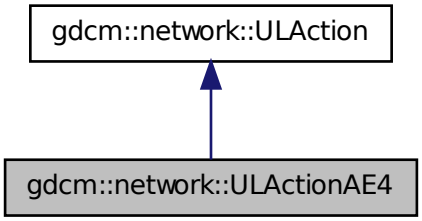
25.281 gdcm::network::ULActionAE4 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE4:



Collaboration diagram for gdcm::network::ULActionAE4:



Public Member Functions

- EStateID PerformAction (Subject ∗s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.281.1 Member Function Documentation

25.281.1.1 EStateID gdcm::network::ULActionAE4::PerformAction (Subject ∗ *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

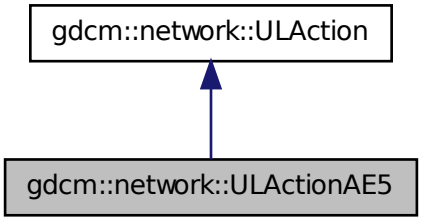
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

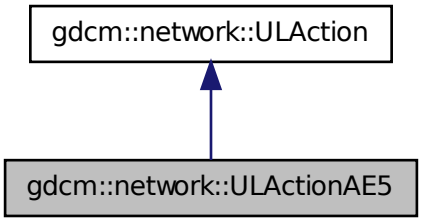
25.282 gdcm::network::ULActionAE5 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE5:



Collaboration diagram for gdcm::network::ULActionAE5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.282.1 Member Function Documentation

25.282.1.1 EStateID gdcm::network::ULActionAE5::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

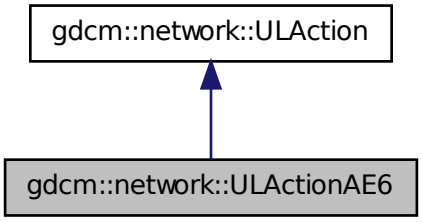
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

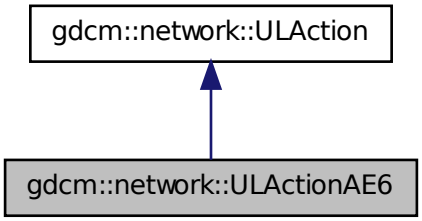
25.283 gdcm::network::ULActionAE6 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE6:



Collaboration diagram for gdcm::network::ULActionAE6:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor-Event, EEventID &outRaisedEvent)

25.283.1 Member Function Documentation

25.283.1.1 EStateID gdcm::network::ULActionAE6::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

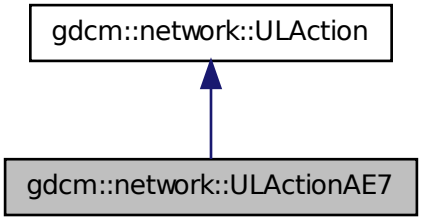
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

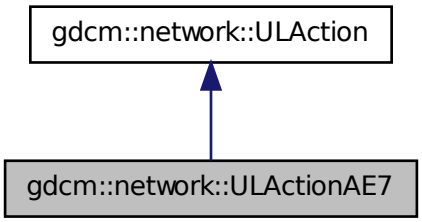
25.284 gdcm::network::ULActionAE7 Class Reference

`#include <gdcmULActionAE.h>`

Inheritance diagram for gdcm::network::ULActionAE7:



Collaboration diagram for gdcm::network::ULActionAE7:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.284.1 Member Function Documentation

25.284.1.1 EStateID gdcm::network::ULActionAE7::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

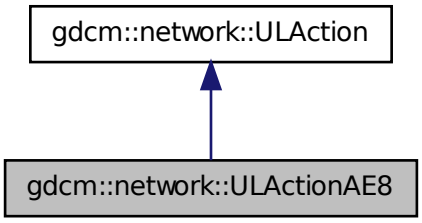
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

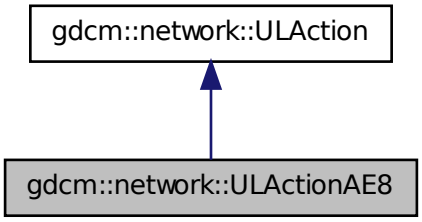
25.285 gdcm::network::ULActionAE8 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE8:



Collaboration diagram for gdcm::network::ULActionAE8:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.285.1 Member Function Documentation

25.285.1.1 EStateID gdcm::network::ULActionAE8::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

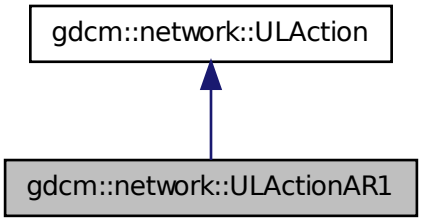
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

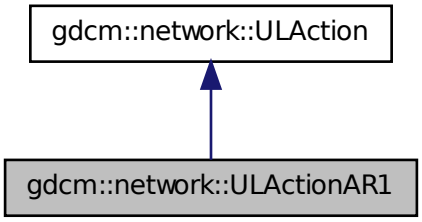
25.286 gdcm::network::ULActionAR1 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR1:



Collaboration diagram for gdcm::network::ULActionAR1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.286.1 Member Function Documentation

25.286.1.1 EStateID gdcm::network::ULActionAR1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

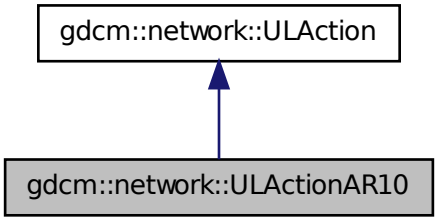
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

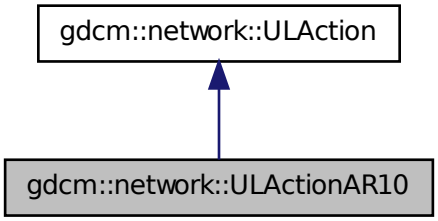
25.287 gdcm::network::ULActionAR10 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR10:



Collaboration diagram for gdcm::network::ULActionAR10:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.287.1 Member Function Documentation

25.287.1.1 EStateID gdcm::network::ULActionAR10::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

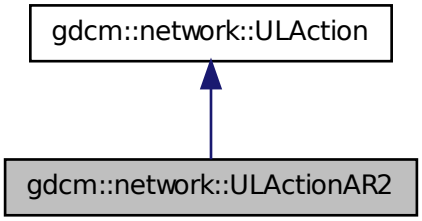
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

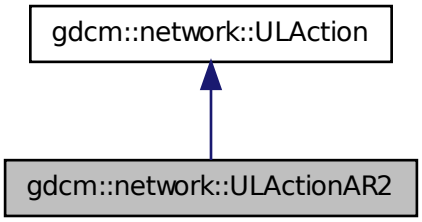
25.288 gdcm::network::ULActionAR2 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR2:



Collaboration diagram for gdcm::network::ULActionAR2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.288.1 Member Function Documentation

25.288.1.1 EStateID gdcm::network::ULActionAR2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

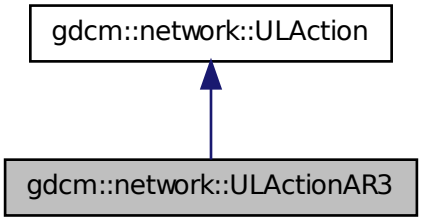
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

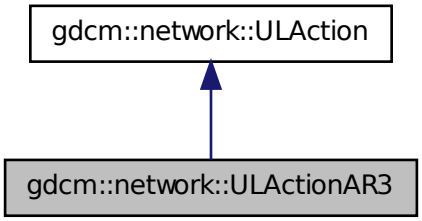
25.289 gdcm::network::ULActionAR3 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR3:



Collaboration diagram for gdcm::network::ULActionAR3:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor-Event, EEventID &outRaisedEvent)

25.289.1 Member Function Documentation

25.289.1.1 EStateID gdcm::network::ULActionAR3::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

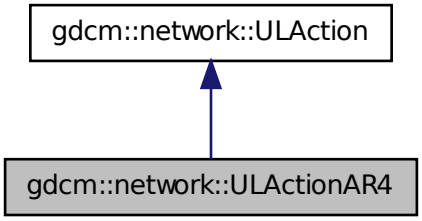
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

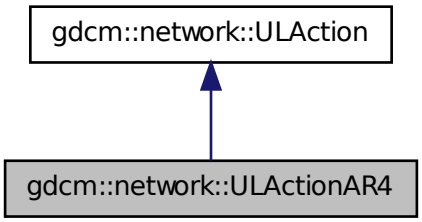
25.290 gdcm::network::ULActionAR4 Class Reference

`#include <gdcmULActionAR.h>`

Inheritance diagram for gdcm::network::ULActionAR4:



Collaboration diagram for gdcm::network::ULActionAR4:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.290.1 Member Function Documentation

25.290.1.1 EStateID gdcm::network::ULActionAR4::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

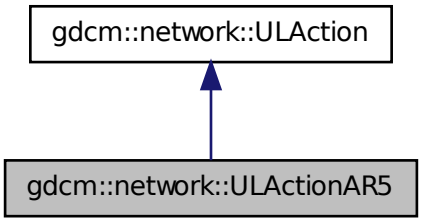
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

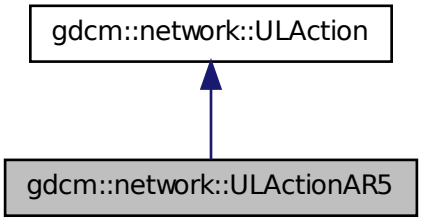
25.291 gdcm::network::ULActionAR5 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR5:



Collaboration diagram for gdcm::network::ULActionAR5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.291.1 Member Function Documentation

25.291.1.1 EStateID gdcm::network::ULActionAR5::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

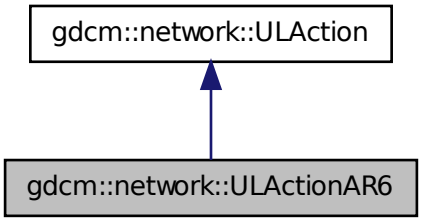
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

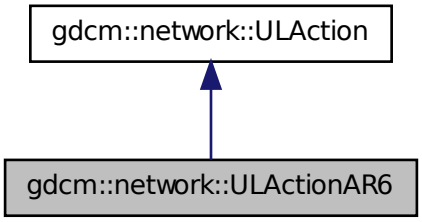
25.292 gdcm::network::ULActionAR6 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR6:



Collaboration diagram for gdcm::network::ULActionAR6:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.292.1 Member Function Documentation

25.292.1.1 EStateID gdcm::network::ULActionAR6::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

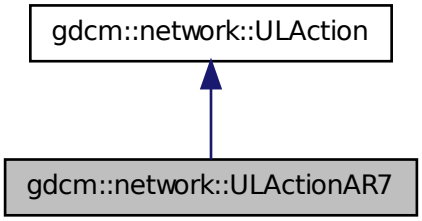
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

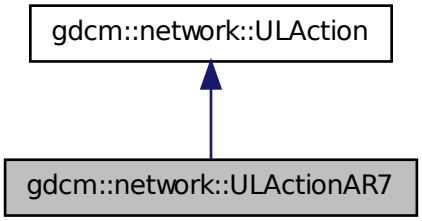
25.293 gdcm::network::ULActionAR7 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR7:



Collaboration diagram for gdcm::network::ULActionAR7:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.293.1 Member Function Documentation

25.293.1.1 EStateID gdcm::network::ULActionAR7::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

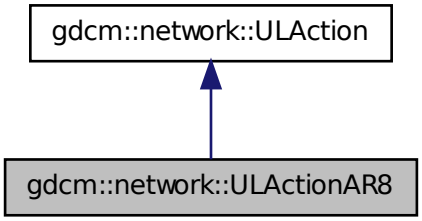
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

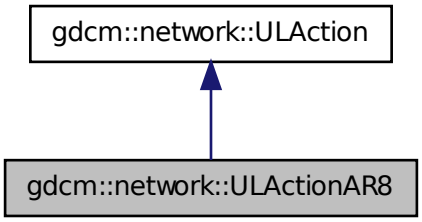
25.294 gdcm::network::ULActionAR8 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR8:



Collaboration diagram for gdcm::network::ULActionAR8:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.294.1 Member Function Documentation

25.294.1.1 EStateID gdcm::network::ULActionAR8::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

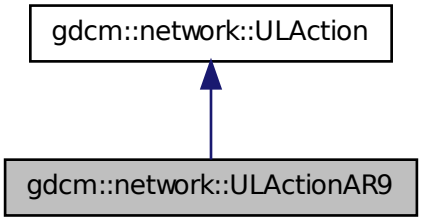
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

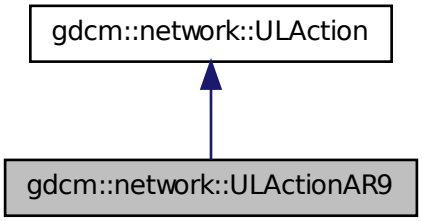
25.295 gdcm::network::ULActionAR9 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR9:



Collaboration diagram for gdcm::network::ULActionAR9:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor-Event, EEventID &outRaisedEvent)

25.295.1 Member Function Documentation

25.295.1.1 EStateID gdcm::network::ULActionAR9::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

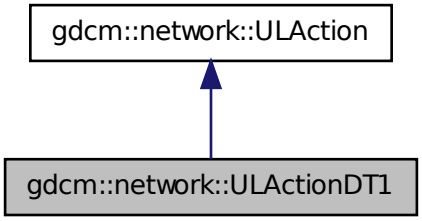
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

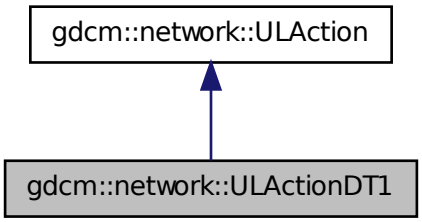
25.296 gdcm::network::ULActionDT1 Class Reference

`#include <gdcmULActionDT.h>`

Inheritance diagram for gdcm::network::ULActionDT1:



Collaboration diagram for gdcm::network::ULActionDT1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.296.1 Member Function Documentation

25.296.1.1 EStateID gdcm::network::ULActionDT1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

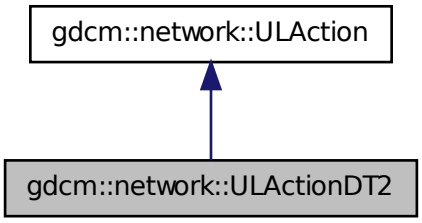
The documentation for this class was generated from the following file:

- gdcmULActionDT.h

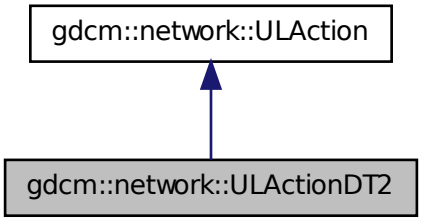
25.297 gdcm::network::ULActionDT2 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for gdcm::network::ULActionDT2:



Collaboration diagram for gdcm::network::ULActionDT2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.297.1 Member Function Documentation

25.297.1.1 EStateID gdcm::network::ULActionDT2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

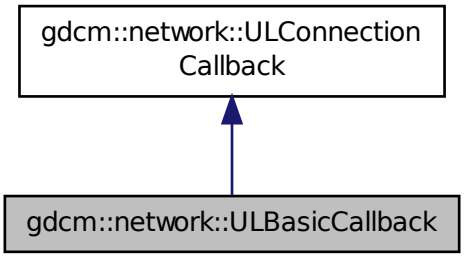
The documentation for this class was generated from the following file:

- gdcmULActionDT.h

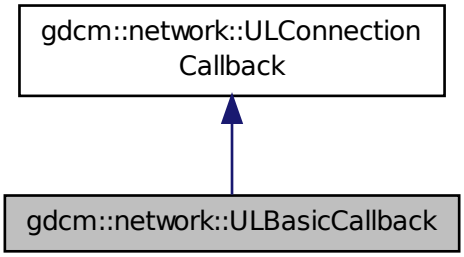
25.298 gdcm::network::ULBasicCallback Class Reference

```
#include <gdcmULBasicCallback.h>
```

Inheritance diagram for gdcm::network::ULBasicCallback:



Collaboration diagram for gdcm::network::ULBasicCallback:



Public Member Functions

- ULBasicCallback ()
- virtual ∼ULBasicCallback ()
- std::vector< DataSet > const & GetDataSets () const
- virtual void HandleDataSet (const DataSet &inDataSet)

Additional Inherited Members

25.298.1 Detailed Description

This is the most basic of callbacks for how the ULConnectionManager handles ∗incoming datasets. DataSets are just concatenated to the mDataSets vector, ∗and the result can be pulled out of the vector by later code. ∗Alternatives to this method include progress updates, saving to disk, etc. ∗This class is NOT THREAD SAFE. Access the dataset vector after the ∗entire set of datasets has been returned by the ULConnectionManager.

25.298.2 Constructor & Destructor Documentation

25.298.2.1 gdcm::network::ULBasicCallback::ULBasicCallback ( ) [inline]

25.298.2.2 virtual gdcm::network::ULBasicCallback::∼ULBasicCallback ( ) [inline],[virtual]

25.298.3 Member Function Documentation

25.298.3.1 std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetDataSets ( ) const

25.298.3.2 virtual void gdcm::network::ULBasicCallback::HandleDataSet (const DataSet & *inDataSet*) [virtual]

Implements gdcm::network::ULConnectionCallback.

The documentation for this class was generated from the following file:

- gdcmULBasicCallback.h

25.299 gdcm::network::ULConnection Class Reference

ULConnection This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

```
#include <gdcmULConnection.h>
```

Public Member Functions

- ULConnection (const ULConnectionInfo &inUserInformation)
- virtual ∼ULConnection ()
- void AddAcceptedPresentationContext (const PresentationContextAC &inPC)
- PresentationContextRQ FindContext (const DataElement &de) const
- std::vector
  < PresentationContextAC >
  const & GetAcceptedPresentationContexts () const
- std::vector
  < PresentationContextAC > & GetAcceptedPresentationContexts ()
- const ULConnectionInfo & GetConnectionInfo () const
- uint32_t GetMaxPDUSize () const
- const PresentationContextAC ∗ GetPresentationContextACByID (uint8_t id) const
- uint8_t GetPresentationContextIDFromPresentationContext (PresentationContextRQ const &pc) const

  return 0 upon error
- const PresentationContextRQ ∗ GetPresentationContextRQByID (uint8_t id) const
- std::vector
  < PresentationContextRQ >
  const & GetPresentationContexts () const
- std::iostream ∗ GetProtocol ()
- EStateID GetState () const
- ARTIMTimer & GetTimer ()
- bool InitializeConnection ()

  used to establish scu connections
- bool InitializeIncomingConnection ()

  used to establish scp connections
- void SetMaxPDUSize (uint32_t inSize)
- void SetPresentationContexts (const std::vector< PresentationContextRQ > &inContexts)
- void SetPresentationContexts (const std::vector< PresentationContext > &inContexts)
- void SetState (const EStateID &inState)
- void StopProtocol ()

25.299.1 Detailed Description

ULConnection This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

The ULConnectionManager tells the ULConnection what data can actually be sent.

This class is done this way so that it can be eventually be replaced with a ULSecureConnection, if such a protocol is warranted, so that all data that passes through can be managed through a secure connection. For now, this class provides a simple pass-through mechanism to the socket itself.

So, for instance, a gdcm object will be passes to this object, and it will then get passed along the connection, if that connection is in the proper state to do so.

For right now, this class is not directly intended to be inherited from, but the potential for future ULSecureConnection warrants the addition, rather than having everything be managed from within the ULConnectionManager (or this class) without a wrapper.

25.299.2 Constructor & Destructor Documentation

25.299.2.1 gdcm::network::ULConnection::ULConnection (const ULConnectionInfo & *inUserInformation*)

25.299.2.2 virtual gdcm::network::ULConnection::~ULConnection ( ) [virtual]

25.299.3 Member Function Documentation

25.299.3.1 void gdcm::network::ULConnection::AddAcceptedPresentationContext (const PresentationContextAC & *inPC*)

25.299.3.2 PresentationContextRQ gdcm::network::ULConnection::FindContext (const DataElement & *de*) const

25.299.3.3 std::vector<PresentationContextAC> const& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( ) const

25.299.3.4 std::vector<PresentationContextAC>& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( )

25.299.3.5 const ULConnectionInfo& gdcm::network::ULConnection::GetConnectionInfo ( ) const

25.299.3.6 uint32_t gdcm::network::ULConnection::GetMaxPDUSize ( ) const

25.299.3.7 const PresentationContextAC* gdcm::network::ULConnection::GetPresentationContextACByID (uint8_t *id*) const

25.299.3.8 uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (PresentationContextRQ const & *pc*) const

return 0 upon error

25.299.3.9 const PresentationContextRQ* gdcm::network::ULConnection::GetPresentationContextRQByID (uint8_t *id*) const

25.299.3.10 std::vector<PresentationContextRQ> const& gdcm::network::ULConnection::GetPresentationContexts ( ) const

25.299.3.11 std::iostream* gdcm::network::ULConnection::GetProtocol ( )

25.299.3.12 EStateID gdcm::network::ULConnection::GetState ( ) const

25.299.3.13 ARTIMTimer& gdcm::network::ULConnection::GetTimer ( )

25.299.3.14 bool gdcm::network::ULConnection::InitializeConnection ( )

used to establish scu connections

25.299.3.15 bool gdcm::network::ULConnection::InitializeIncomingConnection ( )

used to establish scp connections

25.299.3.16 void gdcm::network::ULConnection::SetMaxPDUSize (uint32_t *inSize*)

25.299.3.17 void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< PresentationContextRQ > & *inContexts*)

25.299.3.18 void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< PresentationContext > & *inContexts*)

25.299.3.19 void gdcm::network::ULConnection::SetState (const EStateID & *inState*)

25.299.3.20 void gdcm::network::ULConnection::StopProtocol ( )

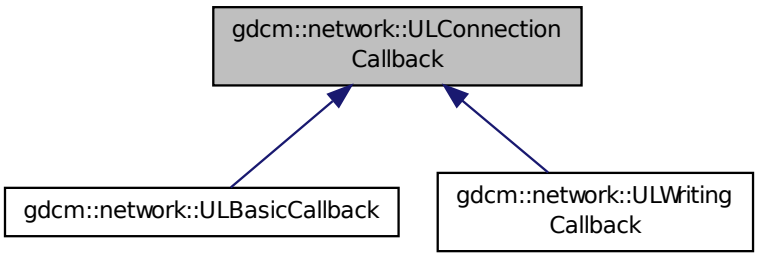
The documentation for this class was generated from the following file:

- gdcmULConnection.h

25.300 gdcm::network::ULConnectionCallback Class Reference

```
#include <gdcmULConnectionCallback.h>
```

Inheritance diagram for gdcm::network::ULConnectionCallback:



Public Member Functions

- ULConnectionCallback ()
- ~ULConnectionCallback ()
- bool DataSetHandles () const
- virtual void HandleDataSet (const DataSet &inDataSet)=0
- void ResetHandledDataSet ()

Protected Member Functions

- void DataSetHandled ()

25.300.1 Detailed Description

When a dataset comes back from a query/move/etc, the result can either be ∗stored entirely in memory, or could be stored on disk. This class provides ∗a mechanism to indicate what the ULConnectionManager should do with datasets ∗that are produced through query results. ∗The ULConnectionManager will call the HandleDataSet function during the course ∗of receiving datasets. Particular implementations should fill in what that ∗function does, including updating progress, etc. ∗NOTE: since cmove requires that multiple event loops be employed, ∗the callback function MUST set mHandledDataSet to true. ∗otherwise, the cmove event loop handler will not know data was received, and ∗proceed to end the loop prematurely.

25.300.2 Constructor & Destructor Documentation

25.300.2.1 gdcm::network::ULConnectionCallback::ULConnectionCallback ( ) [inline]

25.300.2.2 gdcm::network::ULConnectionCallback::∼ULConnectionCallback ( ) [inline]

25.300.3 Member Function Documentation

25.300.3.1 void gdcm::network::ULConnectionCallback::DataSetHandled ( ) [inline],[protected]

25.300.3.2 bool gdcm::network::ULConnectionCallback::DataSetHandles ( ) const [inline]

25.300.3.3 virtual void gdcm::network::ULConnectionCallback::HandleDataSet (const DataSet & *inDataSet*) [pure virtual]

Implemented in gdcm::network::ULWritingCallback, and gdcm::network::ULBasicCallback.

25.300.3.4 void gdcm::network::ULConnectionCallback::ResetHandledDataSet ( ) [inline]

The documentation for this class was generated from the following file:

- gdcmULConnectionCallback.h

25.301 gdcm::network::ULConnectionInfo Class Reference

ULConnectionInfo this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

```
#include <gdcmULConnectionInfo.h>
```

Public Member Functions

- ULConnectionInfo ()
- const char ∗ GetCalledAETitle () const

- std::string GetCalledComputerName () const
- unsigned long GetCalledIPAddress () const
- int GetCalledIPPort () const
- const char ∗ GetCallingAETitle () const
- unsigned long GetMaxPDULength () const
- UserInformation GetUserInformation () const
- bool Initialize (UserInformation inUserInformation, const char inCalledAETitle[16], const char inCallingAETitle[16], unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void SetMaxPDULength (unsigned long inMaxPDULength)

25.301.1 Detailed Description

ULConnectionInfo this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

25.301.2 Constructor & Destructor Documentation

25.301.2.1 gdcm::network::ULConnectionInfo::ULConnectionInfo ( )

25.301.3 Member Function Documentation

25.301.3.1 const char∗ gdcm::network::ULConnectionInfo::GetCalledAETitle ( ) const

25.301.3.2 std::string gdcm::network::ULConnectionInfo::GetCalledComputerName ( ) const

25.301.3.3 unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress ( ) const

25.301.3.4 int gdcm::network::ULConnectionInfo::GetCalledIPPort ( ) const

25.301.3.5 const char∗ gdcm::network::ULConnectionInfo::GetCallingAETitle ( ) const

25.301.3.6 unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength ( ) const

25.301.3.7 UserInformation gdcm::network::ULConnectionInfo::GetUserInformation ( ) const

25.301.3.8 bool gdcm::network::ULConnectionInfo::Initialize (UserInformation *inUserInformation,* const char *inCalledAETitle[16],* const char *inCallingAETitle[16],* unsigned long *inCalledIPAddress,* int *inCalledIPPort,* std::string *inCalledComputerName*)

25.301.3.9 void gdcm::network::ULConnectionInfo::SetMaxPDULength (unsigned long *inMaxPDULength*)

The documentation for this class was generated from the following file:

- gdcmULConnectionInfo.h

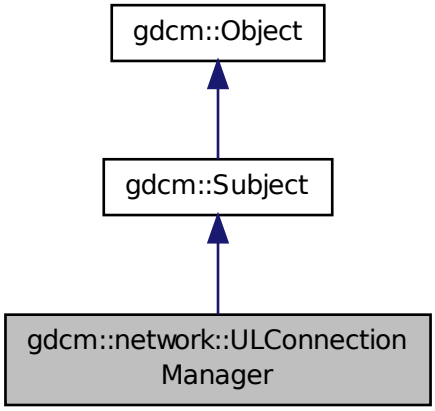
25.302 gdcm::network::ULConnectionManager Class Reference

ULConnectionManager The ULConnectionManager performs actions on the ULConnection given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer

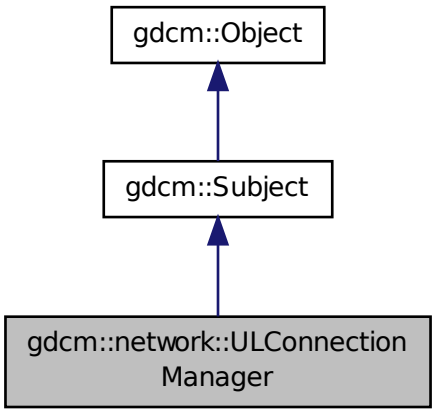
across the connection, etc).

```
#include <gdcmULConnectionManager.h>
```

Inheritance diagram for gdcm::network::ULConnectionManager:



Collaboration diagram for gdcm::network::ULConnectionManager:



Public Member Functions

- ULConnectionManager ()

- ~ULConnectionManager ()
- bool BreakConnection (const double &inTimeout)
- void BreakConnectionNow ()
- bool EstablishConnection (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, std::vector< PresentationContext > const &pcVector)
- bool EstablishConnectionMove (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, uint16_t inReturnPort, std::vector< PresentationContext > const &pcVector)
- std::vector < PresentationDataValue > SendEcho ()
- std::vector< DataSet > SendFind (const BaseRootQuery *inRootQuery)
- void SendFind (const BaseRootQuery *inRootQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendMove (const BaseRootQuery *inRootQuery)
- void SendMove (const BaseRootQuery *inRootQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendStore (const File &file)
- void SendStore (const File &file, ULConnectionCallback *inCallback)

  callback based API

25.302.1 Detailed Description

ULConnectionManager The ULConnectionManager performs actions on the ULConnection given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

Its inputs are ULEvents, and it performs ULActions.

25.302.2 Constructor & Destructor Documentation

25.302.2.1 gdcm::network::ULConnectionManager::ULConnectionManager ( )

25.302.2.2 gdcm::network::ULConnectionManager::~ULConnectionManager ( )

25.302.3 Member Function Documentation

25.302.3.1 bool gdcm::network::ULConnectionManager::BreakConnection (const double & *inTimeout*)

25.302.3.2 void gdcm::network::ULConnectionManager::BreakConnectionNow ( )

25.302.3.3 bool gdcm::network::ULConnectionManager::EstablishConnection (const std::string & *inAETitle,* const std::string & *inConnectAETitle,* const std::string & *inComputerName,* long *inIPAddress,* uint16_t *inConnectPort,* double *inTimeout,* std::vector< **PresentationContext** > const & *pcVector*)

returns true if a connection of the given AETitle (ie, 'this' program) is able to connect to the given AETitle and Port in a certain amount of time providing the connection type will establish the proper exchange syntax with a server; if a different functionality is required, a different connection should be established. returns false if the connection type is 'move'– have to give a return port for move to work as specified.

25.302.3.4 bool gdcm::network::ULConnectionManager::EstablishConnectionMove (const std::string & *inAETitle,* const std::string & *inConnectAETitle,* const std::string & *inComputerName,* long *inIPAddress,* uint16_t *inConnectPort,* double *inTimeout,* uint16_t *inReturnPort,* std::vector< **PresentationContext** > const & *pcVector*)

returns true for above reasons, but contains the special 'move' port

25.302.3.5 std::vector< **PresentationDataValue** > gdcm::network::ULConnectionManager::SendEcho ( )

25.302.3.6 std::vector< **DataSet** > gdcm::network::ULConnectionManager::SendFind (const **BaseRootQuery** * *inRootQuery*)

25.302.3.7 void gdcm::network::ULConnectionManager::SendFind (const **BaseRootQuery** * *inRootQuery,* **ULConnectionCallback** * *inCallback*)

25.302.3.8 std::vector< **DataSet** > gdcm::network::ULConnectionManager::SendMove (const **BaseRootQuery** * *inRootQuery*)

25.302.3.9 void gdcm::network::ULConnectionManager::SendMove (const **BaseRootQuery** * *inRootQuery,* **ULConnectionCallback** * *inCallback*)

25.302.3.10 std::vector< **DataSet** > gdcm::network::ULConnectionManager::SendStore (const **File** & *file*)

25.302.3.11 void gdcm::network::ULConnectionManager::SendStore (const **File** & *file,* **ULConnectionCallback** * *inCallback*)

callback based API

The documentation for this class was generated from the following file:

- gdcmULConnectionManager.h

25.303 gdcm::network::ULEvent Class Reference

ULEvent base class for network events.

```
#include <gdcmULEvent.h>
```

Public Member Functions

- ULEvent (const EEventID &inEventID, std::vector< BasePDU * > inBasePDU)
- ULEvent (const EEventID &inEventID, BasePDU *inBasePDU)
- ~ULEvent ()
- EEventID GetEvent () const
- std::vector< BasePDU * > GetPDUs () const
- void SetEvent (const EEventID &inEvent)
- void SetPDU (std::vector< BasePDU * > inPDU)

25.303.1 Detailed Description

ULEvent base class for network events.

An event consists of the event ID and the data associated with that event.

Note that once a PDU is created, it is now the responsibility of the associated event to destroy it!

25.303.2 Constructor & Destructor Documentation

25.303.2.1 gdcm::network::ULEvent::ULEvent (const EEventID & *inEventID,* std::vector< BasePDU * > *inBasePDU*) [inline]

25.303.2.2 gdcm::network::ULEvent::ULEvent (const EEventID & *inEventID,* BasePDU * *inBasePDU*) [inline]

25.303.2.3 gdcm::network::ULEvent::~ULEvent ( ) [inline]

25.303.3 Member Function Documentation

25.303.3.1 EEventID gdcm::network::ULEvent::GetEvent ( ) const [inline]

25.303.3.2 std::vector<BasePDU*> gdcm::network::ULEvent::GetPDUs ( ) const [inline]

25.303.3.3 void gdcm::network::ULEvent::SetEvent (const EEventID & *inEvent*) [inline]

25.303.3.4 void gdcm::network::ULEvent::SetPDU (std::vector< BasePDU * > *inPDU*) [inline]

The documentation for this class was generated from the following file:

- gdcmULEvent.h

25.304 gdcm::network::ULTransitionTable Class Reference

ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

```
#include <gdcmULTransitionTable.h>
```

Public Member Functions

- ULTransitionTable ()
- void HandleEvent (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) const
- void PrintTable () const

25.304.1 Detailed Description

ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

Based roughly on the solutions in player2.cpp in the boost examples and this so question: http://stackoverflow.com/questions/1647631/c-state-machine-design

The transition table is constructed of TableRows. Each row is based on an event, and an event handler in the TransitionTable object takes a given event, and then finds the given row.

Then, given the current state of the connection, determines the appropriate action to take and then the state to transition to next.

25.304.2 Constructor & Destructor Documentation

25.304.2.1 gdcm::network::ULTransitionTable::ULTransitionTable ( )

25.304.3 Member Function Documentation

25.304.3.1 void gdcm::network::ULTransitionTable::HandleEvent (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) const

25.304.3.2 void gdcm::network::ULTransitionTable::PrintTable ( ) const

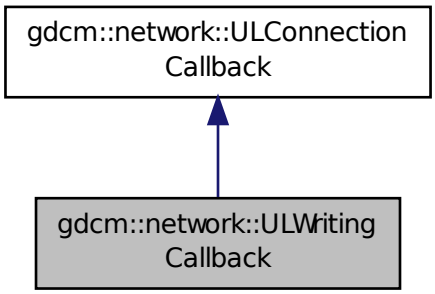
The documentation for this class was generated from the following file:

- gdcmULTransitionTable.h

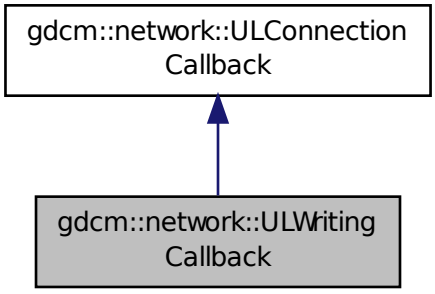
25.305 gdcm::network::ULWritingCallback Class Reference

```
#include <gdcmULWritingCallback.h>
```

Inheritance diagram for gdcm::network::ULWritingCallback:



Collaboration diagram for gdcm::network::ULWritingCallback:



Public Member Functions

- ULWritingCallback ()
- virtual ~ULWritingCallback ()
- virtual void HandleDataSet (const DataSet &inDataSet)
- void SetDirectory (const std::string &inDirectoryName)

  provide the directory into which all files are written.

Additional Inherited Members

25.305.1 Constructor & Destructor Documentation

25.305.1.1 gdcm::network::ULWritingCallback::ULWritingCallback ( ) [inline]

25.305.1.2 virtual gdcm::network::ULWritingCallback::~ULWritingCallback ( ) [inline], [virtual]

25.305.2 Member Function Documentation

25.305.2.1 virtual void gdcm::network::ULWritingCallback::HandleDataSet (const DataSet & *inDataSet*) [virtual]

Implements gdcm::network::ULConnectionCallback.

25.305.2.2 void gdcm::network::ULWritingCallback::SetDirectory (const std::string & *inDirectoryName*) [inline]

provide the directory into which all files are written.

The documentation for this class was generated from the following file:

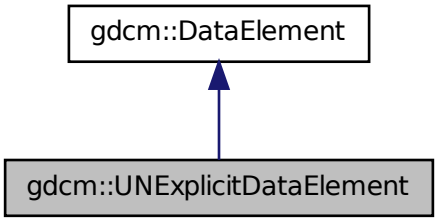
- gdcmULWritingCallback.h

25.306 gdcm::UNExplicitDataElement Class Reference

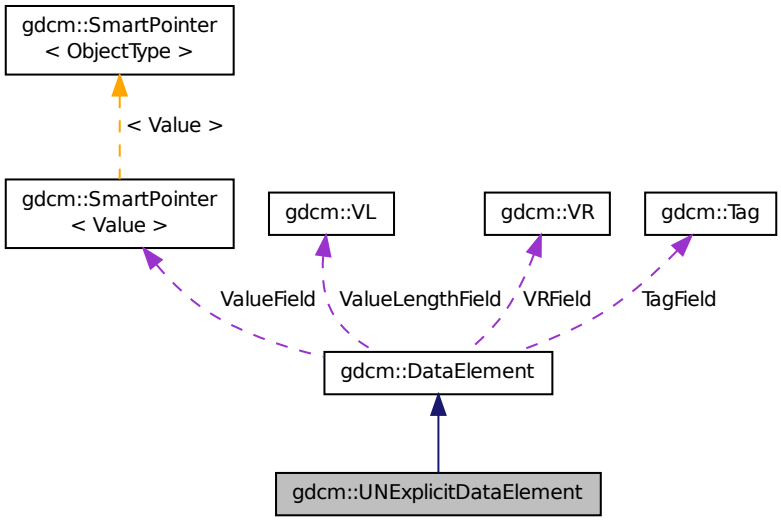
Class to read/write a DataElement as UNExplicit Data Element.

```
#include <gdcmUNExplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitDataElement:



Collaboration diagram for gdcm::UNExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)

- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)

Additional Inherited Members

25.306.1 Detailed Description

Class to read/write a DataElement as UNExplicit Data Element.

Note

bla

25.306.2 Member Function Documentation

25.306.2.1 VL gdcm::UNExplicitDataElement::GetLength ( ) const

Reimplemented from gdcm::DataElement.

25.306.2.2 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::Read (std::istream & *is*)

Reimplemented from gdcm::DataElement.

25.306.2.3 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadPreValue (std::istream & *is*)

25.306.2.4 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadValue (std::istream & *is*)

25.306.2.5 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*)

Reimplemented from gdcm::DataElement.

The documentation for this class was generated from the following file:

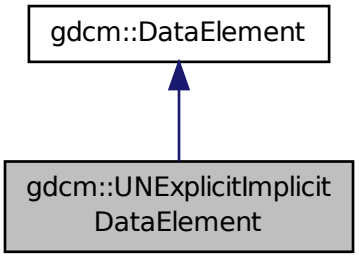
- gdcmUNExplicitDataElement.h

25.307 gdcm::UNExplicitImplicitDataElement Class Reference

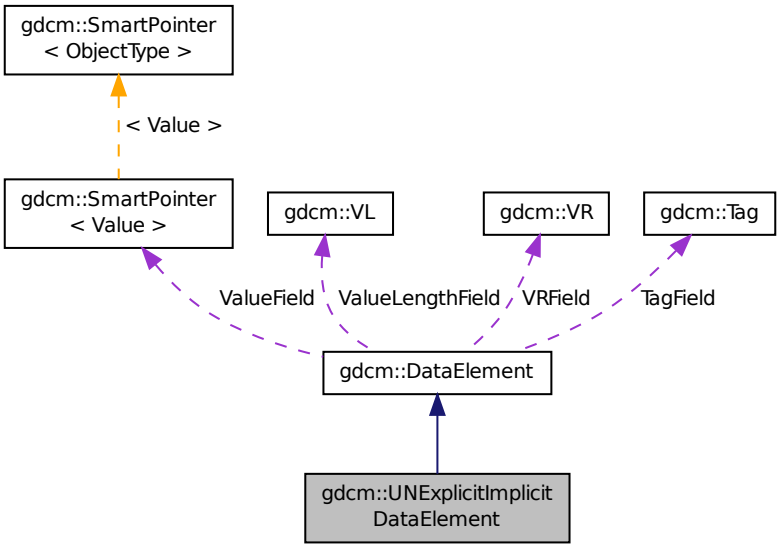
Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:

```
#include <gdcmUNExplicitImplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitImplicitDataElement:



Collaboration diagram for gdcm::UNExplicitImplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)

Additional Inherited Members

25.307.1 Detailed Description

Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:

1. GDCM 1.2.0 would rewrite VR=UN Value Length on 2 bytes instead of 4 bytes
2. GDCM 1.2.0 would also rewrite DataElement as Implicit when the VR would not be known this would only happen in some very rare cases. gdcm 2.X design could handle bug #1 or #2 exclusively, this class can now handle file which have both issues. See: gdcmData/TheralysGDCM120Bug.dcm

25.307.2 Member Function Documentation

25.307.2.1 VL gdcm::UNExplicitImplicitDataElement::GetLength ( ) const

Reimplemented from gdcm::DataElement.

25.307.2.2 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::Read (std::istream & *is*)

Reimplemented from gdcm::DataElement.

25.307.2.3 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadPreValue (std::istream & *is*)

25.307.2.4 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadValue (std::istream & *is*)

The documentation for this class was generated from the following file:

- gdcmUNExplicitImplicitDataElement.h

25.308 gdcm::Unpacker12Bits Class Reference

Pack/Unpack 12 bits pixel into 16bits.

```
#include <gdcmUnpacker12Bits.h>
```

Static Public Member Functions

- static bool Pack (char *out, const char *in, size_t n)
- static bool Unpack (char *out, const char *in, size_t n)

25.308.1 Detailed Description

Pack/Unpack 12 bits pixel into 16bits.

- You can only pack an even number of 16bits, which means a multiple of 4 (expressed in bytes)
- You can only unpack a multiple of 3 bytes

This class has no purpose in general purpose DICOM implementation. However to be able to cope with some early ACR-NEMA file generated by a well-known private vendor, one would need to unpack 12bits Stored Pixel Value into a more standard 16bits Stored Pixel Value.

See also

Rescaler

25.308.2 Member Function Documentation

25.308.2.1 static bool gdcm::Unpacker12Bits::Pack (char * *out*, const char * *in*, size_t *n*) [static]

Pack an array of 16bits where all values are 12bits into a pack form. n is the length in bytes of array in, out will be a fake 8bits array of size (n / 2) * 3

25.308.2.2 static bool gdcm::Unpacker12Bits::Unpack (char * *out*, const char * *in*, size_t *n*) [static]

Unpack an array of 'packed' 12bits data into a more conventional 16bits array. n is the length in bytes of array in, out will be a 16bits array of size (n / 3) * 2

The documentation for this class was generated from the following file:

- gdcmUnpacker12Bits.h

25.309 gdcm::Usage Class Reference

Usage.

```
#include <gdcmUsage.h>
```

Public Types

- enum UsageType {
  Mandatory,
  Conditional,
  UserOption,
  Invalid }

Public Member Functions

- Usage (UsageType type=Invalid)
- operator UsageType () const

Static Public Member Functions

- static const char * GetUsageString (UsageType type)
- static UsageType GetUsageType (const char *type)

Friends

- std::ostream & operator<< (std::ostream &os, const Usage &vr)

25.309.1 Detailed Description

Usage.

Note

A.1.3 IOD Module Table and Functional Group Macro Table This Section of each IOD defines in a tabular form the Modules comprising the IOD. The following information must be specified for each Module in the table:

- The name of the Module or Functional Group
- A reference to the Section in Annex C which defines the Module or Functional Group
- The usage of the Module or Functional Group; whether it is:
- Mandatory (see A.1.3.1) , abbreviated M
- Conditional (see A.1.3.2) , abbreviated C
- User Option (see A.1.3.3) , abbreviated U The Modules referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each IOD, Mandatory Modules shall be supported per the definitions, semantics and requirements defined in Annex C.

A.1.3.2 CONDITIONAL MODULES Conditional Modules are Mandatory Modules if specific conditions are met. If the specified conditions are not met, this Module shall not be supported; that is, no information defined in that Module shall be sent. A.1.3.3 USER OPTION MODULES User Option Modules may or may not be supported. If an optional Module is supported, the Attribute Types specified in the Modules in Annex C shall be supported.

25.309.2 Member Enumeration Documentation

25.309.2.1 enum gdcm::Usage::UsageType

Enumerator:

Mandatory

Conditional

UserOption

Invalid

25.309.3 Constructor & Destructor Documentation

25.309.3.1 gdcm::Usage::Usage (UsageType *type* = Invalid) [inline]

25.309.4 Member Function Documentation

25.309.4.1 static const char* gdcm::Usage::GetUsageString (UsageType *type*) [static]

Referenced by gdcm::operator<<().

25.309.4.2 static UsageType gdcm::Usage::GetUsageType (const char * *type*) [static]

25.309.4.3 gdcm::Usage::operator UsageType ( ) const [inline]

25.309.5 Friends And Related Function Documentation

25.309.5.1 std::ostream& operator<< (std::ostream & *os*, const Usage & *vr*) [friend]

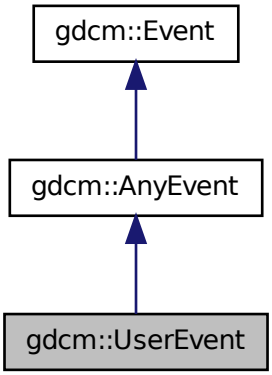
The documentation for this class was generated from the following file:

- gdcmUsage.h

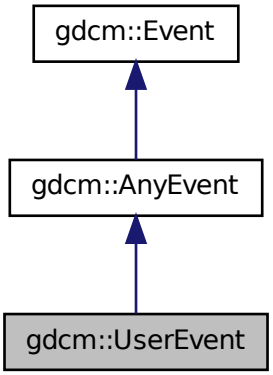
25.310 gdcm::UserEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::UserEvent:



Collaboration diagram for gdcm::UserEvent:



The documentation for this class was generated from the following file:

- gdcmEvent.h

25.311 gdcm::network::UserInformation Class Reference

UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

```
#include <gdcmUserInformation.h>
```

Public Member Functions

- UserInformation ()
- const MaximumLengthSub & GetMaximumLengthSub () const
- MaximumLengthSub & GetMaximumLengthSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.311.1 Detailed Description

UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

TODO what is the goal of :

Table 9-20 USER INFORMATION ITEM FIELDS

25.311.2 Constructor & Destructor Documentation

25.311.2.1 gdcm::network::UserInformation::UserInformation ( )

25.311.3 Member Function Documentation

25.311.3.1 const MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ( ) const [inline]

25.311.3.2 MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ( ) [inline]

25.311.3.3 void gdcm::network::UserInformation::Print (std::ostream & *os*) const

25.311.3.4 std::istream& gdcm::network::UserInformation::Read (std::istream & *is*)

25.311.3.5 size_t gdcm::network::UserInformation::Size ( ) const

25.311.3.6 const std::ostream& gdcm::network::UserInformation::Write (std::ostream & *os*) const

The documentation for this class was generated from the following file:

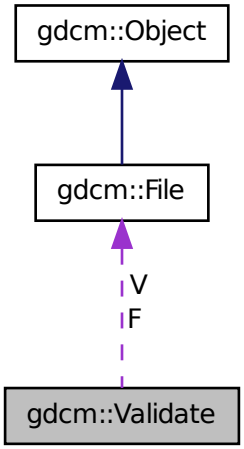
- gdcmUserInformation.h

25.312 gdcm::Validate Class Reference

Validate class.

```
#include <gdcmValidate.h>
```

Collaboration diagram for gdcm::Validate:



Public Member Functions

- Validate ()
- ∼Validate ()
- const File & GetValidatedFile ()
- void SetFile (File const &f)
- void Validation ()

Protected Attributes

- const File ∗ F
- File V

25.312.1 Detailed Description

Validate class.

25.312.2 Constructor & Destructor Documentation

25.312.2.1 gdcm::Validate::Validate ( )

25.312.2.2 gdcm::Validate::∼Validate ( )

25.312.3 Member Function Documentation

25.312.3.1 const File& gdcm::Validate::GetValidatedFile ( ) [inline]

25.312.3.2 void gdcm::Validate::SetFile (File const & *f*) [inline]

25.312.3.3 void gdcm::Validate::Validation ( )

25.312.4 Member Data Documentation

25.312.4.1 const File∗ gdcm::Validate::F [protected]

25.312.4.2 File gdcm::Validate::V [protected]

The documentation for this class was generated from the following file:

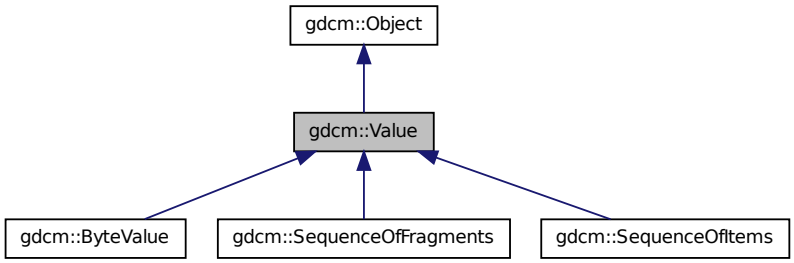
- gdcmValidate.h

25.313 gdcm::Value Class Reference

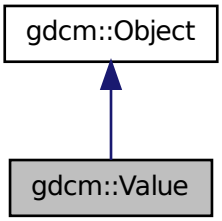
Class to represent the value of a Data Element.

```
#include <gdcmValue.h>
```

Inheritance diagram for gdcm::Value:



Collaboration diagram for gdcm::Value:



Public Member Functions

- Value ()
- ~Value ()
- virtual void Clear ()=0
- virtual VL GetLength () const =0
- virtual bool operator== (const Value &val) const =0
- virtual void SetLength (VL l)=0

Additional Inherited Members

25.313.1 Detailed Description

Class to represent the value of a Data Element.

Note

VALUE: A component of a Value Field. A Value Field may consist of one or more of these components.

25.313.2 Constructor & Destructor Documentation

25.313.2.1 gdcm::Value::Value ( ) [inline]

25.313.2.2 gdcm::Value::~Value ( ) [inline]

25.313.3 Member Function Documentation

25.313.3.1 virtual void gdcm::Value::Clear ( ) [pure virtual]

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

25.313.3.2 virtual VL gdcm::Value::GetLength ( ) const [pure virtual]

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

Referenced by gdcm::DataSet::InsertDataElement(), and gdcm::DataElement::SetValue().

25.313.3.3 virtual bool gdcm::Value::operator== (const Value & *val*) const [pure virtual]

Implemented in gdcm::SequenceOfItems, gdcm::SequenceOfFragments, and gdcm::ByteValue.

25.313.3.4 virtual void gdcm::Value::SetLength (VL *l*) [pure virtual]

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

The documentation for this class was generated from the following file:

- gdcmValue.h

25.314 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference

Class to dispatch template calls.

```
#include <gdcmValueIO.h>
```

Static Public Member Functions

- static std::istream & Read (std::istream &is, Value &v)
- static const std::ostream & Write (std::ostream &os, const Value &v)

25.314.1 Detailed Description

template<typename TDE, typename TSwap, typename TType = uint8_t>class gdcm::ValueIO< TDE, TSwap, TType >

Class to dispatch template calls.

25.314.2 Member Function Documentation

25.314.2.1 template<typename TDE , typename TSwap , typename TType = uint8_t> static std::istream& gdcm::ValueIO< TDE, TSwap, TType >::Read (std::istream & *is*, Value & *v*) [static]

25.314.2.2 template<typename TDE , typename TSwap , typename TType = uint8_t> static const std::ostream& gdcm::ValueIO< TDE, TSwap, TType >::Write (std::ostream & *os*, const Value & *v*) [static]

The documentation for this class was generated from the following file:

- gdcmValueIO.h

25.315 gdcm::Version Class Reference

major/minor and build version

```
#include <gdcmVersion.h>
```

Public Member Functions

- Version ()
- ~Version ()
- void Print (std::ostream &os=std::cout) const

Static Public Member Functions

- static int GetBuildVersion ()
- static int GetMajorVersion ()
- static int GetMinorVersion ()
- static const char * GetVersion ()

Friends

- std::ostream & operator<< (std::ostream &_os, const Version &v)

25.315.1 Detailed Description

major/minor and build version

25.315.2 Constructor & Destructor Documentation

25.315.2.1 gdcm::Version::Version ( ) [inline]

25.315.2.2 gdcm::Version::~Version ( ) [inline]

25.315.3 Member Function Documentation

25.315.3.1 static int gdcm::Version::GetBuildVersion ( ) [static]

25.315.3.2 static int gdcm::Version::GetMajorVersion ( ) [static]

25.315.3.3 static int gdcm::Version::GetMinorVersion ( ) [static]

25.315.3.4 static const char* gdcm::Version::GetVersion ( ) [static]

25.315.3.5 void gdcm::Version::Print (std::ostream & *os* = `std::cout`) const

Referenced by gdcm::operator<<().

25.315.4 Friends And Related Function Documentation

25.315.4.1 std::ostream& operator<< (std::ostream & *_os*, const Version & *v*) [friend]

The documentation for this class was generated from the following file:

- gdcmVersion.h

25.316 gdcm::VL Class Reference

Value Length.

`#include <gdcmVL.h>`

Public Types

- typedef uint32_t Type

Public Member Functions

- VL (uint32_t vl=0)
- VL GetLength () const
- bool IsOdd () const

  Return whether or not the VL is odd or not.
- bool IsUndefined () const
- operator uint32_t () const
- VL & operator++ ()
- VL operator++ (int)
- VL & operator+= (VL const &vl)

  += operator
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & Read16 (std::istream &is)
- void SetToUndefined ()
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

- template<typename TSwap >
  const std::ostream & Write16 (std::ostream &os) const

Static Public Member Functions

- static uint16_t GetVL16Max ()
- static uint32_t GetVL32Max ()

Friends

- std::ostream & operator<< (std::ostream &os, const VL &vl)

25.316.1 Detailed Description

Value Length.

Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

25.316.2 Member Typedef Documentation

25.316.2.1 typedef uint32_t gdcm::VL::Type

25.316.3 Constructor & Destructor Documentation

25.316.3.1 gdcm::VL::VL (uint32_t *vl* = 0) [inline]

25.316.4 Member Function Documentation

25.316.4.1 VL gdcm::VL::GetLength ( ) const [inline]

Referenced by gdcm::FileMetaInformation::GetFullLength(), gdcm::Fragment::GetLength(), and gdcm::Item::Write().

25.316.4.2 static uint16_t gdcm::VL::GetVL16Max ( ) [inline], [static]

25.316.4.3 static uint32_t gdcm::VL::GetVL32Max ( ) [inline], [static]

25.316.4.4 bool gdcm::VL::IsOdd ( ) const [inline]

Return whether or not the VL is odd or not.

Referenced by gdcm::ByteValue::SetLength().

25.316.4.5 bool gdcm::VL::IsUndefined ( ) const [inline]

Referenced by gdcm::ByteValue::SetLength().

25.316.4.6 gdcm::VL::operator uint32_t ( ) const [inline]

25.316.4.7 VL& gdcm::VL::operator++ ( ) [inline]

25.316.4.8 VL gdcm::VL::operator++ (int) [inline]

25.316.4.9 VL& gdcm::VL::operator+= (VL const & *vl*) [inline]

+= operator

25.316.4.10 template<typename TSwap > std::istream& gdcm::VL::Read (std::istream & *is*) [inline]

Referenced by gdcm::Fragment::Read().

25.316.4.11 template<typename TSwap > std::istream& gdcm::VL::Read16 (std::istream & *is*) [inline]

25.316.4.12 void gdcm::VL::SetToUndefined ( ) [inline]

25.316.4.13 template<typename TSwap > const std::ostream& gdcm::VL::Write (std::ostream & *os*) const [inline]

Referenced by gdcm::Fragment::Write(), gdcm::SequenceOfFragments::Write(), gdcm::SequenceOfItems::Write(), and gdcm::Item::Write().

25.316.4.14 template<typename TSwap > const std::ostream& gdcm::VL::Write16 (std::ostream & *os*) const [inline]

25.316.5 Friends And Related Function Documentation

25.316.5.1 std::ostream& operator<< (std::ostream & *os*, const VL & *vl*) [friend]

The documentation for this class was generated from the following file:

- gdcmVL.h

25.317 gdcm::VM Class Reference

Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

```
#include <gdcmVM.h>
```

Public Types

- enum VMType {
  VM0 = 0,
  VM1 = 1,
  VM2 = 2,
  VM3 = 4,
  VM4 = 8,
  VM5 = 16,
  VM6 = 32,
  VM8 = 64,
  VM9 = 128,
  VM10 = 256,
  VM12 = 512,
  VM16 = 1024,
  VM18 = 2048,
  VM24 = 4096,
  VM28 = 8192,
  VM32 = 16384,
  VM35 = 32768,
  VM99 = 65536,
  VM256 = 131072,
  VM1_2 = VM1 | VM2,
  VM1_3 = VM1 | VM2 | VM3,
  VM1_4 = VM1 | VM2 | VM3 | VM4,
  VM1_5 = VM1 | VM2 | VM3 | VM4 | VM5,
  VM1_8 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8,
  VM1_32 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32,
  VM1_99 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99,
  VM1_n = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
  VM2_2n = VM2 | VM4 | VM6 | VM8 | VM16 | VM24 | VM32 | VM256,
  VM2_n = VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
  VM3_4 = VM3 | VM4,
  VM3_3n = VM3 | VM6 | VM9 | VM24 | VM99 | VM256,
  VM3_n = VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
  VM4_4n = VM4 | VM16 | VM24 | VM32 | VM256,
  VM6_6n = VM6 | VM12 | VM18 | VM24,
  VM7_7n,
  VM30_30n,
  VM47_47n,
  VM_END = VM1_n + 1 }

Public Member Functions

- VM (VMType type=VM0)
- bool Compatible (VM const &vm) const
- unsigned int GetLength () const
- operator VMType () const

Static Public Member Functions

- static unsigned int GetNumberOfElementsFromArray (const char *array, unsigned int length)

- static const char ∗ GetVMString (VMType vm)
- static VMType GetVMType (const char ∗vm)
- static VMType GetVMTypeFromLength (unsigned int length, unsigned int size)
- static bool IsValid (int vm1, VMType vm2)

Static Protected Member Functions

- static unsigned int GetIndex (VMType vm)

Friends

- std::ostream & operator<< (std::ostream &os, const VM &vm)

25.317.1 Detailed Description

Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

Some private dict define some more: 4-4n 1-4 1-5 256 9 3-4

even more:

7-7n 10 18 12 35 47_47n 30_30n 28

6-6n

25.317.2 Member Enumeration Documentation

25.317.2.1 enum gdcm::VM::VMType

Enumerator:

VM0

VM1

VM2

VM3

VM4

VM5

VM6

VM8

VM9

VM10

VM12

VM16

VM18

VM24

VM28

VM32

VM35

VM99

VM256

VM1_2

VM1_3

VM1_4

VM1_5

VM1_8

VM1_32

VM1_99

VM1_n

VM2_2n

VM2_n

VM3_4

VM3_3n

VM3_n

VM4_4n

VM6_6n

VM7_7n

VM30_30n

VM47_47n

VM_END

25.317.3 Constructor & Destructor Documentation

25.317.3.1 gdcm::VM::VM (VMType *type* = VM0) [inline]

25.317.4 Member Function Documentation

25.317.4.1 bool gdcm::VM::Compatible (VM const & *vm*) const

WARNING: Implementation deficiency The Compatible function is poorly implemented, the reference vm should be coming from the dictionary, while the passed in value is the value guess from the file.

25.317.4.2 static unsigned int gdcm::VM::GetIndex (VMType *vm*) [static], [protected]

25.317.4.3 unsigned int gdcm::VM::GetLength ( ) const

25.317.4.4 static unsigned int gdcm::VM::GetNumberOfElementsFromArray (const char * *array,* unsigned int *length*) [static]

25.317.4.5 static const char* gdcm::VM::GetVMString (VMType *vm*) [static]

Return the string as written in the official DICOM dict from a custom enum type

Referenced by gdcm::operator<<().

25.317.4.6 static VMType gdcm::VM::GetVMType (const char * *vm*) [static]

25.317.4.7 static VMType gdcm::VM::GetVMTypeFromLength (unsigned int *length,* unsigned int *size*) [static]

25.317.4.8 static bool gdcm::VM::IsValid (int *vm1,* VMType *vm2*) [static]

Check if vm1 is valid compare to vm2, i.e vm1 is element of vm2 vm1 is typically deduce from counting in a ValueField

25.317.4.9 gdcm::VM::operator VMType ( ) const [inline]

25.317.5 Friends And Related Function Documentation

25.317.5.1 std::ostream& operator<< (std::ostream & *os,* const VM & *vm*) [friend]

The documentation for this class was generated from the following file:

- gdcmVM.h

25.318 gdcm::VR Class Reference

VR class This is adapted from DICOM standard The biggest difference is the INVALID VR and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict.

```
#include <gdcmVR.h>
```

Public Types

- enum VRType {
  INVALID = 0,
  AE = 1,
  AS = 2,
  AT = 4,
  CS = 8,
  DA = 16,
  DS = 32,
  DT = 64,
  FD = 128,
  FL = 256,
  IS = 512,
  LO = 1024,
  LT = 2048,
  OB = 4096,
  OF = 8192,
  OW = 16384,
  PN = 32768,
  SH = 65536,
  SL = 131072,
  SQ = 262144,
  SS = 524288,
  ST = 1048576,
  TM = 2097152,
  UI = 4194304,
  UL = 8388608,
  UN = 16777216,
  US = 33554432,
  UT = 67108864,
  OB_OW = OB | OW,
  US_SS = US | SS,
  US_SS_OW = US | SS | OW,
  VL16 = AE | AS | AT | CS | DA | DS | DT | FD | FL | IS | LO | LT | PN | SH | SL | SS | ST | TM | UI | UL | US,
  VL32 = OB | OW | OF | SQ | UN | UT,
  VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UI | UT,
  VRBINARY = AT | FL | FD | OB | OF | OW | SL | SQ | SS | UL | UN | US,
  VR_VM1 = AS | LT | ST | UT | SQ | OF | OW | OB | UN,
  VRALL = VRASCII | VRBINARY,
  VR_END = UT+1 }

Public Member Functions

- VR (VRType vr=INVALID)
- bool Compatible (VR const &vr) const
- int GetLength () const
- unsigned int GetSize () const
- unsigned int GetSizeof () const
- bool IsDual () const
- bool IsVRFile () const
- operator VRType () const
- std::istream & Read (std::istream &is)

- const std::ostream & Write (std::ostream &os) const

Static Public Member Functions

- static bool CanDisplay (VRType vr)
- static uint32_t GetLength (VRType vr)
- static const char ∗ GetVRString (VRType vr)
- static const char ∗ GetVRStringFromFile (VRType vr)
- static VRType GetVRType (const char ∗vr)
- static VRType GetVRTypeFromFile (const char ∗vr)
- static bool IsASCII (VRType vr)
- static bool IsASCII2 (VRType vr)
- static bool IsBinary (VRType vr)
- static bool IsBinary2 (VRType vr)
- static bool IsSwap (const char ∗vr)
- static bool IsValid (const char ∗vr)
- static bool IsValid (const char ∗vr1, VRType vr2)

Friends

- std::ostream & operator<< (std::ostream &os, const VR &vr)

25.318.1 Detailed Description

VR class This is adapted from DICOM standard The biggest difference is the INVALID VR and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict.

Note

VALUE REPRESENTATION (VR) Specifies the data type and format of the Value(s) contained in the Value Field of a Data Element. VALUE REPRESENTATION FIELD: The field where the Value Representation of a Data Element is stored in the encoding of a Data Element structure with explicit VR.

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

25.318.2 Member Enumeration Documentation

25.318.2.1 enum gdcm::VR::VRType

Enumerator:

INVALID

AE

AS

AT

CS

DA

DS
DT
FD
FL
IS
LO
LT
OB
OF
OW
PN
SH
SL
SQ
SS
ST
TM
UI
UL
UN
US
UT
OB_OW
US_SS
US_SS_OW
VL16
VL32
VRASCII
VRBINARY
VR_VM1
VRALL
VR_END

25.318.3 Constructor & Destructor Documentation

25.318.3.1 gdcm::VR::VR (VRType *vr* = INVALID) [inline]

25.318.4 Member Function Documentation

25.318.4.1 static bool gdcm::VR::CanDisplay (VRType *vr*) [static]

25.318.4.2 bool gdcm::VR::Compatible (VR const & *vr*) const

25.318.4.3 int gdcm::VR::GetLength ( ) const [inline]

25.318.4.4 static uint32_t gdcm::VR::GetLength (VRType *vr*) [inline], [static]

25.318.4.5 unsigned int gdcm::VR::GetSize ( ) const [inline]

References AE, US_SS, and VRTypeTemplateCase.

25.318.4.6 unsigned int gdcm::VR::GetSizeof ( ) const

25.318.4.7 static const char* gdcm::VR::GetVRString (VRType *vr*) [static]

Referenced by gdcm::operator<<().

25.318.4.8 static const char* gdcm::VR::GetVRStringFromFile (VRType *vr*) [static]

25.318.4.9 static VRType gdcm::VR::GetVRType (const char * *vr*) [static]

25.318.4.10 static VRType gdcm::VR::GetVRTypeFromFile (const char * *vr*) [static]

25.318.4.11 static bool gdcm::VR::IsASCII (VRType *vr*) [static]

25.318.4.12 static bool gdcm::VR::IsASCII2 (VRType *vr*) [static]

25.318.4.13 static bool gdcm::VR::IsBinary (VRType *vr*) [static]

25.318.4.14 static bool gdcm::VR::IsBinary2 (VRType *vr*) [static]

25.318.4.15 bool gdcm::VR::IsDual ( ) const

25.318.4.16 static bool gdcm::VR::IsSwap (const char * *vr*) [static]

25.318.4.17 static bool gdcm::VR::IsValid (const char * *vr*) [static]

25.318.4.18 static bool gdcm::VR::IsValid (const char * *vr1*, VRType *vr2*) [static]

25.318.4.19 bool gdcm::VR::IsVRFile ( ) const

Referenced by gdcm::DataElement::SetVR().

25.318.4.20 gdcm::VR::operator VRType ( ) const [inline]

25.318.4.21 std::istream& gdcm::VR::Read (std::istream & *is*) [inline]

References gdcmDebugMacro, INVALID, and VR_END.

25.318.4.22 const std::ostream& gdcm::VR::Write (std::ostream & *os*) const [inline]

References gdcmAssertAlwaysMacro, and INVALID.

25.318.5 Friends And Related Function Documentation

25.318.5.1 std::ostream& operator<< (std::ostream & *os*, const VR & *vr*) [friend]

The documentation for this class was generated from the following file:

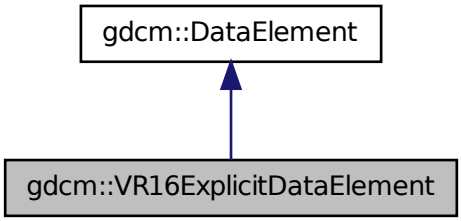
- gdcmVR.h

25.319 gdcm::VR16ExplicitDataElement Class Reference

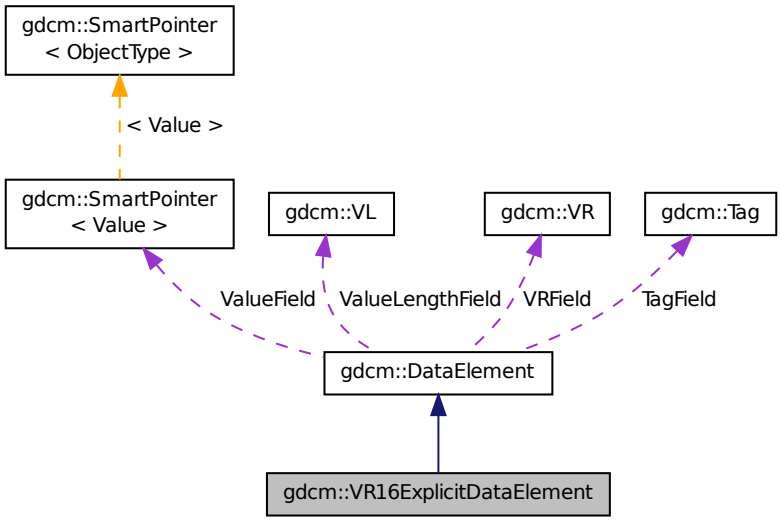
Class to read/write a DataElement as Explicit Data Element.

`#include <gdcmVR16ExplicitDataElement.h>`

Inheritance diagram for gdcm::VR16ExplicitDataElement:



Collaboration diagram for gdcm::VR16ExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)

Additional Inherited Members

25.319.1 Detailed Description

Class to read/write a DataElement as Explicit Data Element.

Note

This class support 16 bits when finding an unkown VR: For instance: Siemens_CT_Sensation64_has_VR_RT.dcm

25.319.2 Member Function Documentation

25.319.2.1 VL gdcm::VR16ExplicitDataElement::GetLength ( ) const

Reimplemented from gdcm::DataElement.

25.319.2.2 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::Read (std::istream & *is*)

Reimplemented from gdcm::DataElement.

25.319.2.3 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadPreValue (std::istream & *is*)

25.319.2.4 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadValue (std::istream & *is*)

25.319.2.5 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*)

Reimplemented from gdcm::DataElement.

The documentation for this class was generated from the following file:

- gdcmVR16ExplicitDataElement.h

25.320 gdcm::VRVLSize< 0 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Static Public Member Functions

- static uint16_t Read (std::istream &_is)
- static void Write (std::ostream &os)

25.320.1 Member Function Documentation

25.320.1.1 static uint16_t gdcm::VRVLSize< 0 >::Read (std::istream & *_is*) [inline], [static]

25.320.1.2 static void gdcm::VRVLSize< 0 >::Write (std::ostream & *os*) [inline], [static]

The documentation for this class was generated from the following file:

- gdcmAttribute.h

25.321 gdcm::VRVLSize< 1 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Static Public Member Functions

- static uint32_t Read (std::istream &_is)
- static void Write (std::ostream &os)

25.321.1 Member Function Documentation

25.321.1.1 static uint32_t gdcm::VRVLSize< 1 >::Read (std::istream & *_is*) [inline], [static]

25.321.1.2 static void gdcm::VRVLSize< 1 >::Write (std::ostream & *os*) [inline], [static]

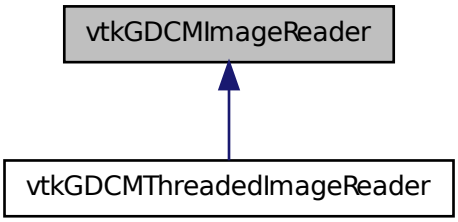
The documentation for this class was generated from the following file:

- gdcmAttribute.h

25.322 vtkGDCMImageReader Class Reference

```
#include <vtkGDCMImageReader.h>
```

Inheritance diagram for vtkGDCMImageReader:



Public Member Functions

- virtual int CanReadFile (const char *fname)
- virtual const char * GetDescriptiveName ()
- virtual const char * GetFileExtensions ()
- vtkImageData * GetIconImage ()
- vtkImageData * GetOverlay (int i)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetCurve (vtkPolyData *pd)
- virtual void SetFileNames (vtkStringArray *)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties *pd)
- vtkBooleanMacro (LoadOverlays, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LoadIconImage, int)
- vtkGetMacro (LossyFlag, int)

- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (LoadIconImage, int)
- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (ApplyLookupTable, int)
- vtkTypeRevisionMacro (vtkGDCMImageReader, vtkMedicalImageReader2)

Static Public Member Functions

- static vtkGDCMImageReader ∗ New ()

Protected Member Functions

- vtkGDCMImageReader ()
- ∼vtkGDCMImageReader ()
- void ExecuteData (vtkDataObject ∗out)
- void ExecuteInformation ()
- void FillMedicalImageInformation (const gdcm::ImageReader &reader)
- int LoadSingleFile (const char ∗filename, char ∗pointer, unsigned long &outlen)
- int RequestDataCompat ()
- int RequestInformationCompat ()
- void SetFilePattern (const char ∗)
- void SetFilePrefix (const char ∗)
- vtkGetStringMacro (FilePrefix)
- vtkGetStringMacro (FilePattern)
- vtkSetVector6Macro (ImageOrientationPatient, double)

Protected Attributes

- int ApplyInverseVideo
- int ApplyLookupTable
- int ApplyPlanarConfiguration
- int ApplyShiftScale
- int ApplyYBRToRGB
- vtkPolyData ∗ Curve
- vtkMatrix4x4 ∗ DirectionCosines

- vtkStringArray * FileNames
- int ForceRescale
- int IconDataScalarType
- int IconImageDataExtent [6]
- int IconNumberOfScalarComponents
- int ImageFormat
- double ImageOrientationPatient [6]
- double ImagePositionPatient [3]
- int LoadIconImage
- int LoadOverlays
- int LossyFlag
- vtkMedicalImageProperties * MedicalImageProperties
- int NumberOfIconImages
- int NumberOfOverlays
- int PlanarConfiguration
- double Scale
- double Shift

25.322.1 Detailed Description

Examples:

AWTMedical3.java, Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, HelloActiviz.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, HelloVTKWorld.java, MagnifyFile.cxx, MetaImageMD5Activiz.cs, MIPViewer.java, MPRViewer.java, MPRViewer2.java, offscreenimage.cxx, ReadSeriesIntoVTK.java, RefCounting.cs, and reslicesphere.cxx.

25.322.2 Constructor & Destructor Documentation

25.322.2.1 vtkGDCMImageReader::vtkGDCMImageReader ( ) [protected]

25.322.2.2 vtkGDCMImageReader::~vtkGDCMImageReader ( ) [protected]

25.322.3 Member Function Documentation

25.322.3.1 virtual int vtkGDCMImageReader::CanReadFile (const char * *fname*) [virtual]

Examples:

MetaImageMD5Activiz.cs.

25.322.3.2 void vtkGDCMImageReader::ExecuteData (vtkDataObject * *out*) [protected]

Reimplemented in vtkGDCMThreadedImageReader.

25.322.3.3 void vtkGDCMImageReader::ExecuteInformation ( ) [protected]

Reimplemented in vtkGDCMThreadedImageReader.

25.322.3.4 void vtkGDCMImageReader::FillMedicalImageInformation (const gdcm::ImageReader & *reader*) [protected]

25.322.3.5 virtual const char* vtkGDCMImageReader::GetDescriptiveName ( ) [inline], [virtual]

25.322.3.6 virtual const char* vtkGDCMImageReader::GetFileExtensions ( ) [inline], [virtual]

25.322.3.7 vtkImageData* vtkGDCMImageReader::GetIconImage ( )

25.322.3.8 vtkImageData* vtkGDCMImageReader::GetOverlay (int *i*)

25.322.3.9 int vtkGDCMImageReader::LoadSingleFile (const char * *filename,* char * *pointer,* unsigned long & *outlen*) [protected]

25.322.3.10 static vtkGDCMImageReader* vtkGDCMImageReader::New ( ) [static]

Reimplemented in vtkGDCMThreadedImageReader.

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, HelloActiviz.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, MagnifyFile.cxx, MetaImageMD5Activiz.cs, offscreenimage.cxx, RefCounting.cs, and reslicesphere.cxx.

25.322.3.11 virtual void vtkGDCMImageReader::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

Reimplemented in vtkGDCMThreadedImageReader.

25.322.3.12 int vtkGDCMImageReader::RequestDataCompat ( ) [protected]

Reimplemented in vtkGDCMThreadedImageReader.

25.322.3.13 int vtkGDCMImageReader::RequestInformationCompat ( ) [protected]

25.322.3.14 virtual void vtkGDCMImageReader::SetCurve (vtkPolyData * *pd*) [virtual]

25.322.3.15 virtual void vtkGDCMImageReader::SetFileNames (vtkStringArray *) [virtual]

Examples:

gdcmorthoplanes.cxx, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, MIPViewer.java, MPRViewer.java, MPRViewer2.java, and ReadSeriesIntoVTK.java.

25.322.3.16 void vtkGDCMImageReader::SetFilePattern (const char *) [inline], [protected]

25.322.3.17 void vtkGDCMImageReader::SetFilePrefix (const char *) [inline], [protected]

25.322.3.18 virtual void vtkGDCMImageReader::SetMedicalImageProperties (vtkMedicalImageProperties * *pd*) [virtual]

25.322.3.19 vtkGDCMImageReader::vtkBooleanMacro (**LoadOverlays** , int)

25.322.3.20 vtkGDCMImageReader::vtkBooleanMacro (**LoadIconImage** , int)

25.322.3.21 vtkGDCMImageReader::vtkBooleanMacro (**LossyFlag** , int)

25.322.3.22 vtkGDCMImageReader::vtkBooleanMacro (**ApplyLookupTable** , int)

25.322.3.23 int vtkGDCMImageReader::vtkBooleanMacro (**ApplyYBRToRGB** , int)

25.322.3.24 vtkGDCMImageReader::vtkGetMacro (**LoadOverlays** , int)

25.322.3.25 vtkGDCMImageReader::vtkGetMacro (**LoadIconImage** , int)

25.322.3.26 vtkGDCMImageReader::vtkGetMacro (**LossyFlag** , int)

25.322.3.27 vtkGDCMImageReader::vtkGetMacro (**NumberOfOverlays** , int)

25.322.3.28 vtkGDCMImageReader::vtkGetMacro (**NumberOfIconImages** , int)

25.322.3.29 vtkGDCMImageReader::vtkGetMacro (**ApplyLookupTable** , int)

25.322.3.30 vtkGDCMImageReader::vtkGetMacro (**ApplyYBRToRGB** , int)

25.322.3.31 vtkGDCMImageReader::vtkGetMacro (**ImageFormat** , int)

25.322.3.32 vtkGDCMImageReader::vtkGetMacro (**PlanarConfiguration** , int)

25.322.3.33 vtkGDCMImageReader::vtkGetMacro (**Shift** , double)

25.322.3.34 vtkGDCMImageReader::vtkGetMacro (**Scale** , double)

25.322.3.35 vtkGDCMImageReader::vtkGetObjectMacro (**DirectionCosines** , vtkMatrix4x4)

25.322.3.36 vtkGDCMImageReader::vtkGetObjectMacro (**MedicalImageProperties** , vtkMedicalImageProperties)

25.322.3.37 vtkGDCMImageReader::vtkGetObjectMacro (**FileNames** , vtkStringArray)

25.322.3.38 vtkGDCMImageReader::vtkGetObjectMacro (**Curve** , vtkPolyData)

25.322.3.39 vtkGDCMImageReader::vtkGetStringMacro (FilePrefix) [protected]

25.322.3.40 vtkGDCMImageReader::vtkGetStringMacro (FilePattern) [protected]

25.322.3.41 vtkGDCMImageReader::vtkGetVector3Macro (**ImagePositionPatient** , double)

25.322.3.42 vtkGDCMImageReader::vtkGetVector6Macro (**ImageOrientationPatient** , double)

25.322.3.43 vtkGDCMImageReader::vtkSetMacro (**LoadOverlays** , int)

25.322.3.44 vtkGDCMImageReader::vtkSetMacro (**LoadIconImage** , int)

25.322.3.45 vtkGDCMImageReader::vtkSetMacro (**LossyFlag** , int)

25.322.3.46 vtkGDCMImageReader::vtkSetMacro (**ApplyLookupTable** , int)

25.322.3.47 vtkGDCMImageReader::vtkSetVector6Macro (**ImageOrientationPatient** , double) [protected]

25.322.3.48 vtkGDCMImageReader::vtkTypeRevisionMacro (**vtkGDCMImageReader** , vtkMedicalImageReader2)

25.322.4 Member Data Documentation

25.322.4.1 int vtkGDCMImageReader::ApplyInverseVideo [protected]

25.322.4.2 int vtkGDCMImageReader::ApplyLookupTable [protected]

25.322.4.3 int vtkGDCMImageReader::ApplyPlanarConfiguration [protected]

25.322.4.4 int vtkGDCMImageReader::ApplyShiftScale [protected]

25.322.4.5 int vtkGDCMImageReader::ApplyYBRToRGB [protected]

25.322.4.6 vtkPolyData* vtkGDCMImageReader::Curve [protected]

25.322.4.7 vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines [protected]

25.322.4.8 vtkStringArray* vtkGDCMImageReader::FileNames [protected]

25.322.4.9 int vtkGDCMImageReader::ForceRescale [protected]

25.322.4.10 int vtkGDCMImageReader::IconDataScalarType [protected]

25.322.4.11 int vtkGDCMImageReader::IconImageDataExtent[6] [protected]

25.322.4.12 int vtkGDCMImageReader::IconNumberOfScalarComponents [protected]

25.322.4.13 int vtkGDCMImageReader::ImageFormat [protected]

25.322.4.14 double vtkGDCMImageReader::ImageOrientationPatient[6] [protected]

25.322.4.15 double vtkGDCMImageReader::ImagePositionPatient[3] [protected]

25.322.4.16 int vtkGDCMImageReader::LoadIconImage [protected]

25.322.4.17 int vtkGDCMImageReader::LoadOverlays [protected]

25.322.4.18 int vtkGDCMImageReader::LossyFlag [protected]

25.322.4.19 vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties [protected]

25.322.4.20 int vtkGDCMImageReader::NumberOfIconImages [protected]

25.322.4.21 int vtkGDCMImageReader::NumberOfOverlays [protected]

25.322.4.22 int vtkGDCMImageReader::PlanarConfiguration [protected]

25.322.4.23 double vtkGDCMImageReader::Scale [protected]

25.322.4.24 double vtkGDCMImageReader::Shift [protected]

The documentation for this class was generated from the following file:

- vtkGDCMImageReader.h

25.323 vtkGDCMImageWriter Class Reference

```
#include <vtkGDCMImageWriter.h>
```

Public Types

- enum CompressionTypes {
  NO_COMPRESSION = 0,
  JPEG_COMPRESSION,
  JPEG2000_COMPRESSION,
  JPEGLS_COMPRESSION,
  RLE_COMPRESSION }

Public Member Functions

- virtual const char ∗ GetDescriptiveName ()
- virtual const char ∗ GetFileExtensions ()
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetDirectionCosines (vtkMatrix4x4 ∗matrix)
- virtual void SetDirectionCosinesFromImageOrientationPatient (const double dircos[6])
- virtual void SetFileNames (vtkStringArray ∗)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗)
- vtkBooleanMacro (LossyFlag, int)
- vtkBooleanMacro (FileLowerLeft, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (FileLowerLeft, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (CompressionType, int)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetStringMacro (StudyUID)
- vtkGetStringMacro (SeriesUID)
- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (Shift, double)
- vtkSetMacro (Scale, double)

- vtkSetMacro (ImageFormat, int)
- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (PlanarConfiguration, int)
- vtkSetMacro (CompressionType, int)
- vtkSetStringMacro (StudyUID)
- vtkSetStringMacro (SeriesUID)
- vtkTypeRevisionMacro (vtkGDCMImageWriter, vtkImageWriter)
- virtual void Write ()

Static Public Member Functions

- static vtkGDCMImageWriter ∗ New ()

Protected Member Functions

- vtkGDCMImageWriter ()
- ∼vtkGDCMImageWriter ()
- virtual char ∗ GetFileName ()
- int WriteGDCMData (vtkImageData ∗data, int timeStep)
- void WriteSlice (vtkImageData ∗data)

25.323.1 Detailed Description

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, MagnifyFile.cxx, and RefCounting.cs.

25.323.2 Member Enumeration Documentation

25.323.2.1 enum vtkGDCMImageWriter::CompressionTypes

Enumerator:

NO_COMPRESSION

JPEG_COMPRESSION

JPEG2000_COMPRESSION

JPEGLS_COMPRESSION

RLE_COMPRESSION

25.323.3 Constructor & Destructor Documentation

25.323.3.1 vtkGDCMImageWriter::vtkGDCMImageWriter ( ) [protected]

25.323.3.2 vtkGDCMImageWriter::∼vtkGDCMImageWriter ( ) [protected]

25.323.4 Member Function Documentation

25.323.4.1 virtual const char* vtkGDCMImageWriter::GetDescriptiveName ( ) [inline], [virtual]

25.323.4.2 virtual const char* vtkGDCMImageWriter::GetFileExtensions ( ) [inline], [virtual]

25.323.4.3 virtual char* vtkGDCMImageWriter::GetFileName ( ) [protected], [virtual]

25.323.4.4 static vtkGDCMImageWriter* vtkGDCMImageWriter::New ( ) [static]

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloVTKWorld.cs, HelloVTKWorld2.cs, MagnifyFile.cxx, and RefCounting.cs.

25.323.4.5 virtual void vtkGDCMImageWriter::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

25.323.4.6 virtual void vtkGDCMImageWriter::SetDirectionCosines (vtkMatrix4x4 * *matrix*) [virtual]

Examples:

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, and MagnifyFile.cxx.

25.323.4.7 virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (const double *dircos[6]*) [virtual]

25.323.4.8 virtual void vtkGDCMImageWriter::SetFileNames (vtkStringArray *) [virtual]

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.323.4.9 virtual void vtkGDCMImageWriter::SetMedicalImageProperties (vtkMedicalImageProperties *) [virtual]

Examples:

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, and MagnifyFile.cxx.

25.323.4.10 vtkGDCMImageWriter::vtkBooleanMacro (LossyFlag , int)

25.323.4.11 vtkGDCMImageWriter::vtkBooleanMacro (FileLowerLeft , int)

25.323.4.12 vtkGDCMImageWriter::vtkGetMacro (LossyFlag , int)

25.323.4.13 vtkGDCMImageWriter::vtkGetMacro (Shift , double)

25.323.4.14 vtkGDCMImageWriter::vtkGetMacro (Scale , double)

25.323.4.15 vtkGDCMImageWriter::vtkGetMacro (ImageFormat , int)

25.323.4.16 vtkGDCMImageWriter::vtkGetMacro (FileLowerLeft , int)

25.323.4.17 vtkGDCMImageWriter::vtkGetMacro (PlanarConfiguration , int)

25.323.4.18 vtkGDCMImageWriter::vtkGetMacro (CompressionType , int)

25.323.4.19 vtkGDCMImageWriter::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)

25.323.4.20 vtkGDCMImageWriter::vtkGetObjectMacro (FileNames , vtkStringArray)

25.323.4.21 vtkGDCMImageWriter::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)

25.323.4.22 vtkGDCMImageWriter::vtkGetStringMacro (StudyUID)

25.323.4.23 vtkGDCMImageWriter::vtkGetStringMacro (SeriesUID)

25.323.4.24 vtkGDCMImageWriter::vtkSetMacro (LossyFlag , int)

25.323.4.25 vtkGDCMImageWriter::vtkSetMacro (Shift , double)

25.323.4.26 vtkGDCMImageWriter::vtkSetMacro (Scale , double)

25.323.4.27 vtkGDCMImageWriter::vtkSetMacro (ImageFormat , int)

25.323.4.28 vtkGDCMImageWriter::vtkSetMacro (FileLowerLeft , int)

25.323.4.29 vtkGDCMImageWriter::vtkSetMacro (PlanarConfiguration , int)

25.323.4.30 vtkGDCMImageWriter::vtkSetMacro (CompressionType , int)

25.323.4.31 vtkGDCMImageWriter::vtkSetStringMacro (StudyUID)

25.323.4.32 vtkGDCMImageWriter::vtkSetStringMacro (SeriesUID)

25.323.4.33 vtkGDCMImageWriter::vtkTypeRevisionMacro (**vtkGDCMImageWriter** , vtkImageWriter)

25.323.4.34 virtual void vtkGDCMImageWriter::Write ( ) [virtual]

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, and MagnifyFile.cxx.

25.323.4.35 int vtkGDCMImageWriter::WriteGDCMData (vtkImageData * *data,* int *timeStep*) [protected]

25.323.4.36 void vtkGDCMImageWriter::WriteSlice (vtkImageData * *data*) [protected]

The documentation for this class was generated from the following file:

- vtkGDCMImageWriter.h

25.324 vtkGDCMMedicalImageProperties Class Reference

```
#include <vtkGDCMMedicalImageProperties.h>
```

Public Member Functions

- virtual void Clear ()
- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (vtkGDCMMedicalImageProperties, vtkMedicalImageProperties)

Static Public Member Functions

- static vtkGDCMMedicalImageProperties ∗ New ()

Protected Member Functions

- vtkGDCMMedicalImageProperties ()
- ∼vtkGDCMMedicalImageProperties ()
- gdcm::File const & GetFile (unsigned int t)
- void PushBackFile (gdcm::File const &f)

Friends

- class vtkGDCMImageReader
- class vtkGDCMImageWriter

25.324.1 Constructor & Destructor Documentation

25.324.1.1 vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ( ) [protected]

25.324.1.2 vtkGDCMMedicalImageProperties::∼vtkGDCMMedicalImageProperties ( ) [protected]

25.324.2 Member Function Documentation

25.324.2.1 virtual void vtkGDCMMedicalImageProperties::Clear ( ) [virtual]

25.324.2.2 gdcm::File const& vtkGDCMMedicalImageProperties::GetFile (unsigned int *t*) [protected]

25.324.2.3 static vtkGDCMMedicalImageProperties∗ vtkGDCMMedicalImageProperties::New ( ) [static]

25.324.2.4 void vtkGDCMMedicalImageProperties::PrintSelf (ostream & *os*, vtkIndent *indent*)

25.324.2.5 void vtkGDCMMedicalImageProperties::PushBackFile (gdcm::File const & *f*) [protected]

25.324.2.6 vtkGDCMMedicalImageProperties::vtkTypeRevisionMacro (vtkGDCMMedicalImageProperties , vtkMedicalImageProperties)

25.324.3 Friends And Related Function Documentation

25.324.3.1 friend class vtkGDCMImageReader [friend]

25.324.3.2 friend class vtkGDCMImageWriter [friend]

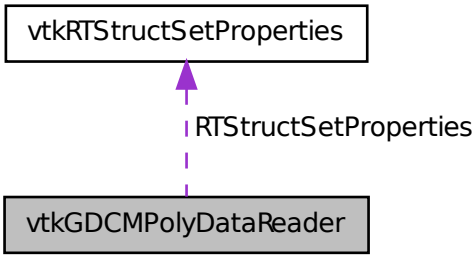
The documentation for this class was generated from the following file:

- vtkGDCMMedicalImageProperties.h

25.325 vtkGDCMPolyDataReader Class Reference

```
#include <vtkGDCMPolyDataReader.h>
```

Collaboration diagram for vtkGDCMPolyDataReader:



Public Member Functions

- virtual void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetObjectMacro (RTStructSetProperties, vtkRTStructSetProperties)
- vtkGetStringMacro (FileName)
- vtkSetStringMacro (FileName)
- vtkTypeRevisionMacro (vtkGDCMPolyDataReader, vtkPolyDataAlgorithm)

Static Public Member Functions

- static vtkGDCMPolyDataReader * New ()

Protected Member Functions

- vtkGDCMPolyDataReader ()
- ∼vtkGDCMPolyDataReader ()
- void FillMedicalImageInformation (const gdcm::Reader &reader)
- int RequestData (vtkInformation ∗, vtkInformationVector ∗∗, vtkInformationVector ∗)
- int RequestData_HemodynamicWaveformStorage (gdcm::Reader const &reader, vtkInformationVector ∗output-Vector)
- int RequestData_RTStructureSetStorage (gdcm::Reader const &reader, vtkInformationVector ∗outputVector)
- int RequestInformation (vtkInformation ∗vtkNotUsed(request), vtkInformationVector ∗∗vtkNotUsed(inputVector), vtkInformationVector ∗outputVector)
- int RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const &reader)
- int RequestInformation_RTStructureSetStorage (gdcm::Reader const &reader)

Protected Attributes

- char ∗ FileName
- vtkMedicalImageProperties ∗ MedicalImageProperties
- vtkRTStructSetProperties ∗ RTStructSetProperties

25.325.1 Detailed Description

Examples:

gdcmscene.cxx, GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.325.2 Constructor & Destructor Documentation

25.325.2.1 vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ( ) [protected]

25.325.2.2 vtkGDCMPolyDataReader::∼vtkGDCMPolyDataReader ( ) [protected]

25.325.3 Member Function Documentation

25.325.3.1 void vtkGDCMPolyDataReader::FillMedicalImageInformation (const gdcm::Reader & *reader*) [protected]

25.325.3.2 static vtkGDCMPolyDataReader∗ vtkGDCMPolyDataReader::New ( ) [static]

Examples:

gdcmscene.cxx, GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.325.3.3 virtual void vtkGDCMPolyDataReader::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

25.325.3.4 int vtkGDCMPolyDataReader::RequestData (vtkInformation ∗ , vtkInformationVector ∗∗ , vtkInformationVector ∗) [protected]

25.325.3.5 int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (gdcm::Reader const & *reader,* vtkInformationVector ∗ *outputVector*) [protected]

25.325.3.6 int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (gdcm::Reader const & *reader,* vtkInformationVector * *outputVector*) [protected]

25.325.3.7 int vtkGDCMPolyDataReader::RequestInformation (vtkInformation * *vtkNotUsedrequest,* vtkInformationVector ** *vtkNotUsedinputVector,* vtkInformationVector * *outputVector*) [protected]

25.325.3.8 int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const & *reader*) [protected]

25.325.3.9 int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (gdcm::Reader const & *reader*) [protected]

25.325.3.10 vtkGDCMPolyDataReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)

25.325.3.11 vtkGDCMPolyDataReader::vtkGetObjectMacro (RTStructSetProperties , vtkRTStructSetProperties)

25.325.3.12 vtkGDCMPolyDataReader::vtkGetStringMacro (FileName)

25.325.3.13 vtkGDCMPolyDataReader::vtkSetStringMacro (FileName)

25.325.3.14 vtkGDCMPolyDataReader::vtkTypeRevisionMacro (vtkGDCMPolyDataReader , vtkPolyDataAlgorithm)

25.325.4 Member Data Documentation

25.325.4.1 char* vtkGDCMPolyDataReader::FileName [protected]

25.325.4.2 vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties [protected]

25.325.4.3 vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties [protected]

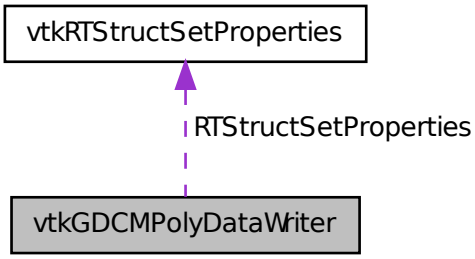
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataReader.h

25.326 vtkGDCMPolyDataWriter Class Reference

```
#include <vtkGDCMPolyDataWriter.h>
```

Collaboration diagram for vtkGDCMPolyDataWriter:



Public Member Functions

- void InitializeRTStructSet (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray ∗inROINames, vtkStringArray ∗inROIAlgorithmName, vtkStringArray ∗inROIType)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗pd)
- void SetNumberOfInputPorts (int n)
- virtual void SetRTStructSetProperties (vtkRTStructSetProperties ∗pd)
- vtkTypeRevisionMacro (vtkGDCMPolyDataWriter, vtkPolyDataWriter)

Static Public Member Functions

- static vtkGDCMPolyDataWriter ∗ New ()

Protected Member Functions

- vtkGDCMPolyDataWriter ()
- ∼vtkGDCMPolyDataWriter ()
- void WriteData ()
- void WriteRTSTRUCTData (gdcm::File &file, int num)
- void WriteRTSTRUCTInfo (gdcm::File &file)

Protected Attributes

- vtkMedicalImageProperties ∗ MedicalImageProperties
- vtkRTStructSetProperties ∗ RTStructSetProperties

25.326.1 Detailed Description

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.326.2 Constructor & Destructor Documentation

25.326.2.1 vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ( ) [protected]

25.326.2.2 vtkGDCMPolyDataWriter::~vtkGDCMPolyDataWriter ( ) [protected]

25.326.3 Member Function Documentation

25.326.3.1 void vtkGDCMPolyDataWriter::InitializeRTStructSet (vtkStdString *inDirectory,* vtkStdString *inStructLabel,* vtkStdString *inStructName,* vtkStringArray * *inROINames,* vtkStringArray * *inROIAlgorithmName,* vtkStringArray * *inROIType*)

Examples:

GenerateRTSTRUCT.cxx.

25.326.3.2 static **vtkGDCMPolyDataWriter*** vtkGDCMPolyDataWriter::New ( ) [static]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.326.3.3 virtual void vtkGDCMPolyDataWriter::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

25.326.3.4 virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (vtkMedicalImageProperties * *pd*) [virtual]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.326.3.5 void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (int *n*)

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.326.3.6 virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (**vtkRTStructSetProperties** * *pd*) [virtual]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.326.3.7 vtkGDCMPolyDataWriter::vtkTypeRevisionMacro (**vtkGDCMPolyDataWriter** , vtkPolyDataWriter)

25.326.3.8 void vtkGDCMPolyDataWriter::WriteData ( ) [protected]

25.326.3.9 void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (**gdcm::File** & *file,* int *num*) [protected]

25.326.3.10 void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (**gdcm::File** & *file*) [protected]

25.326.4 Member Data Documentation

25.326.4.1 vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties [protected]

25.326.4.2 vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties [protected]

The documentation for this class was generated from the following file:

- vtkGDCMPolyDataWriter.h

25.327 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Public Types

- typedef const char *const (* MD5MetaImagesType)[3]

Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (vtkGDCMTesting, vtkObject)

Static Public Member Functions

- static const char * GetGDCMDataRoot ()
- static const char *const * GetMD5MetaImage (unsigned int file)
- static const char * GetMHDMD5FromFile (const char *filepath)
- static unsigned int GetNumberOfMD5MetaImages ()
- static const char * GetRAWMD5FromFile (const char *filepath)
- static const char * GetVTKDataRoot ()
- static vtkGDCMTesting * New ()

Protected Member Functions

- vtkGDCMTesting ()
- ~vtkGDCMTesting ()

25.327.1 Detailed Description

Examples:

HelloActiviz5.cs, HelloVTKWorld2.cs, MetaImageMD5Activiz.cs, ReadSeriesIntoVTK.java, and RefCounting.cs.

25.327.2 Member Typedef Documentation

25.327.2.1 typedef const char* const(* vtkGDCMTesting::MD5MetaImagesType)[3]

25.327.3 Constructor & Destructor Documentation

25.327.3.1 vtkGDCMTesting::vtkGDCMTesting ( ) [protected]

25.327.3.2 vtkGDCMTesting::~vtkGDCMTesting ( ) [protected]

25.327.4 Member Function Documentation

25.327.4.1 static const char* vtkGDCMTesting::GetGDCMDataRoot ( ) [static]

Examples:

HelloActiviz5.cs, and ReadSeriesIntoVTK.java.

25.327.4.2 static const char* const* vtkGDCMTesting::GetMD5MetaImage (unsigned int *file*) [static]

25.327.4.3 static const char* vtkGDCMTesting::GetMHDMD5FromFile (const char * *filepath*) [static]

Examples:

MetaImageMD5Activiz.cs.

25.327.4.4 static unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages ( ) [static]

25.327.4.5 static const char* vtkGDCMTesting::GetRAWMD5FromFile (const char * *filepath*) [static]

Examples:

MetaImageMD5Activiz.cs.

25.327.4.6 static const char* vtkGDCMTesting::GetVTKDataRoot ( ) [static]

Examples:

HelloActiviz5.cs, and HelloVTKWorld2.cs.

25.327.4.7 static vtkGDCMTesting* vtkGDCMTesting::New ( ) [static]

Examples:

RefCounting.cs.

25.327.4.8 void vtkGDCMTesting::PrintSelf (ostream & *os*, vtkIndent *indent*)

25.327.4.9 vtkGDCMTesting::vtkTypeRevisionMacro (**vtkGDCMTesting** , vtkObject)

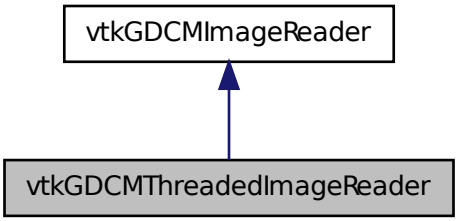
The documentation for this class was generated from the following file:

- vtkGDCMTesting.h

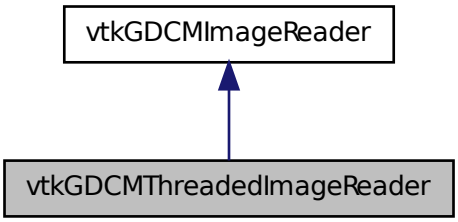
25.328 vtkGDCMThreadedImageReader Class Reference

```
#include <vtkGDCMThreadedImageReader.h>
```

Inheritance diagram for vtkGDCMThreadedImageReader:



Collaboration diagram for vtkGDCMThreadedImageReader:



Public Member Functions

- virtual void PrintSelf (ostream &os, vtkIndent indent)
- vtkBooleanMacro (UseShiftScale, int)
- vtkGetMacro (UseShiftScale, int)

- vtkSetMacro (Shift, double)
- vtkSetMacro (Scale, double)
- vtkSetMacro (UseShiftScale, int)
- vtkTypeRevisionMacro (vtkGDCMThreadedImageReader, vtkGDCMImageReader)

Static Public Member Functions

- static vtkGDCMThreadedImageReader ∗ New ()

Protected Member Functions

- vtkGDCMThreadedImageReader ()
- ∼vtkGDCMThreadedImageReader ()
- void ExecuteData (vtkDataObject ∗out)
- void ExecuteInformation ()
- void ReadFiles (unsigned int nfiles, const char ∗filenames[])
- void RequestDataCompat ()

Additional Inherited Members

25.328.1 Constructor & Destructor Documentation

25.328.1.1 vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ( ) [protected]

25.328.1.2 vtkGDCMThreadedImageReader::∼vtkGDCMThreadedImageReader ( ) [protected]

25.328.2 Member Function Documentation

25.328.2.1 void vtkGDCMThreadedImageReader::ExecuteData (vtkDataObject ∗ *out*) [protected]

Reimplemented from vtkGDCMImageReader.

25.328.2.2 void vtkGDCMThreadedImageReader::ExecuteInformation ( ) [protected]

Reimplemented from vtkGDCMImageReader.

25.328.2.3 static vtkGDCMThreadedImageReader∗ vtkGDCMThreadedImageReader::New ( ) [static]

Reimplemented from vtkGDCMImageReader.

25.328.2.4 virtual void vtkGDCMThreadedImageReader::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

Reimplemented from vtkGDCMImageReader.

25.328.2.5 void vtkGDCMThreadedImageReader::ReadFiles (unsigned int *nfiles,* const char * *filenames[]*) [protected]

25.328.2.6 void vtkGDCMThreadedImageReader::RequestDataCompat ( ) [protected]

Reimplemented from vtkGDCMImageReader.

25.328.2.7 vtkGDCMThreadedImageReader::vtkBooleanMacro (UseShiftScale , int)

25.328.2.8 vtkGDCMThreadedImageReader::vtkGetMacro (UseShiftScale , int)

25.328.2.9 vtkGDCMThreadedImageReader::vtkSetMacro (Shift , double)

25.328.2.10 vtkGDCMThreadedImageReader::vtkSetMacro (Scale , double)

25.328.2.11 vtkGDCMThreadedImageReader::vtkSetMacro (UseShiftScale , int)

25.328.2.12 vtkGDCMThreadedImageReader::vtkTypeRevisionMacro (vtkGDCMThreadedImageReader , vtkGDCMImageReader)

The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader.h

25.329 vtkGDCMThreadedImageReader2 Class Reference

```
#include <vtkGDCMThreadedImageReader2.h>
```

Public Member Functions

- virtual const char * GetFileName (int i=0)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetFileName (const char *filename)
- virtual void SetFileNames (vtkStringArray *)
- int SplitExtent (int splitExt[6], int startExt[6], int num, int total)
- vtkBooleanMacro (FileLowerLeft, int)
- vtkBooleanMacro (LoadOverlays, int)
- vtkBooleanMacro (UseShiftScale, int)
- vtkGetMacro (FileLowerLeft, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (DataScalarType, int)
- vtkGetMacro (NumberOfScalarComponents, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetMacro (UseShiftScale, int)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetVector3Macro (DataOrigin, double)
- vtkGetVector3Macro (DataSpacing, double)
- vtkGetVector6Macro (DataExtent, int)

- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (DataScalarType, int)
- vtkSetMacro (NumberOfScalarComponents, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (Shift, double)
- vtkSetMacro (Scale, double)
- vtkSetMacro (UseShiftScale, int)
- vtkSetVector3Macro (DataOrigin, double)
- vtkSetVector3Macro (DataSpacing, double)
- vtkSetVector6Macro (DataExtent, int)
- vtkTypeRevisionMacro (vtkGDCMThreadedImageReader2, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkGDCMThreadedImageReader2 ∗ New ()

Protected Member Functions

- vtkGDCMThreadedImageReader2 ()
- ∼vtkGDCMThreadedImageReader2 ()
- int RequestInformation (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗outputVector)
- void ThreadedRequestData (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗outputVector, vtkImageData ∗∗∗inData, vtkImageData ∗∗outData, int outExt[6], int id)

25.329.1 Constructor & Destructor Documentation

25.329.1.1 vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ( ) [protected]

25.329.1.2 vtkGDCMThreadedImageReader2::∼vtkGDCMThreadedImageReader2 ( ) [protected]

25.329.2 Member Function Documentation

25.329.2.1 virtual const char∗ vtkGDCMThreadedImageReader2::GetFileName (int *i* = 0) [virtual]

25.329.2.2 static vtkGDCMThreadedImageReader2∗ vtkGDCMThreadedImageReader2::New ( ) [static]

25.329.2.3 virtual void vtkGDCMThreadedImageReader2::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

25.329.2.4 int vtkGDCMThreadedImageReader2::RequestInformation (vtkInformation ∗ *request,* vtkInformationVector ∗∗ *inputVector,* vtkInformationVector ∗ *outputVector*) [protected]

25.329.2.5 virtual void vtkGDCMThreadedImageReader2::SetFileName (const char ∗ *filename*) [virtual]

25.329.2.6 virtual void vtkGDCMThreadedImageReader2::SetFileNames (vtkStringArray ∗) [virtual]

25.329.2.7 int vtkGDCMThreadedImageReader2::SplitExtent (int *splitExt[6],* int *startExt[6],* int *num,* int *total*)

25.329.2.8 void vtkGDCMThreadedImageReader2::ThreadedRequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector,* vtkImageData *** *inData,* vtkImageData ** *outData,* int *outExt[6],* int *id*) [protected]

25.329.2.9 vtkGDCMThreadedImageReader2::vtkBooleanMacro (FileLowerLeft , int)

25.329.2.10 vtkGDCMThreadedImageReader2::vtkBooleanMacro (LoadOverlays , int)

25.329.2.11 vtkGDCMThreadedImageReader2::vtkBooleanMacro (UseShiftScale , int)

25.329.2.12 vtkGDCMThreadedImageReader2::vtkGetMacro (FileLowerLeft , int)

25.329.2.13 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfOverlays , int)

25.329.2.14 vtkGDCMThreadedImageReader2::vtkGetMacro (DataScalarType , int)

25.329.2.15 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfScalarComponents , int)

25.329.2.16 vtkGDCMThreadedImageReader2::vtkGetMacro (LoadOverlays , int)

25.329.2.17 vtkGDCMThreadedImageReader2::vtkGetMacro (Shift , double)

25.329.2.18 vtkGDCMThreadedImageReader2::vtkGetMacro (Scale , double)

25.329.2.19 vtkGDCMThreadedImageReader2::vtkGetMacro (UseShiftScale , int)

25.329.2.20 vtkGDCMThreadedImageReader2::vtkGetObjectMacro (FileNames , vtkStringArray)

25.329.2.21 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataOrigin , double)

25.329.2.22 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataSpacing , double)

25.329.2.23 vtkGDCMThreadedImageReader2::vtkGetVector6Macro (DataExtent , int)

25.329.2.24 vtkGDCMThreadedImageReader2::vtkSetMacro (FileLowerLeft , int)

25.329.2.25 vtkGDCMThreadedImageReader2::vtkSetMacro (DataScalarType , int)

25.329.2.26 vtkGDCMThreadedImageReader2::vtkSetMacro (NumberOfScalarComponents , int)

25.329.2.27 vtkGDCMThreadedImageReader2::vtkSetMacro (LoadOverlays , int)

25.329.2.28 vtkGDCMThreadedImageReader2::vtkSetMacro (Shift , double)

25.329.2.29 vtkGDCMThreadedImageReader2::vtkSetMacro (Scale , double)

25.329.2.30 vtkGDCMThreadedImageReader2::vtkSetMacro (UseShiftScale , int)

25.329.2.31 vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataOrigin , double)

25.329.2.32 vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataSpacing , double)

25.329.2.33 vtkGDCMThreadedImageReader2::vtkSetVector6Macro (DataExtent , int)

25.329.2.34 vtkGDCMThreadedImageReader2::vtkTypeRevisionMacro (vtkGDCMThreadedImageReader2 , vtkThreadedImageAlgorithm)

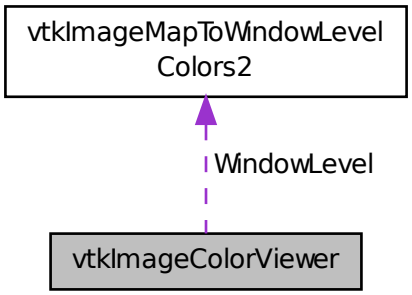
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader2.h

25.330 vtkImageColorViewer Class Reference

```
#include <vtkImageColorViewer.h>
```

Collaboration diagram for vtkImageColorViewer:



Public Types

- enum {
  SLICE_ORIENTATION_YZ = 0,
  SLICE_ORIENTATION_XZ = 1,
  SLICE_ORIENTATION_XY = 2 }

Public Member Functions

- virtual void AddInput (vtkImageData ∗input)
- virtual void AddInputConnection (vtkAlgorithmOutput ∗input)
- virtual double GetColorLevel ()
- virtual double GetColorWindow ()
- virtual vtkImageData ∗ GetInput ()
- virtual int GetOffScreenRendering ()
- double GetOverlayVisibility ()
- virtual int ∗ GetPosition ()
- virtual int ∗ GetSize ()

- virtual int GetSliceMax ()
- virtual int GetSliceMin ()
- virtual void GetSliceRange (int range[2])
- virtual void GetSliceRange (int &min, int &max)
- virtual int ∗ GetSliceRange ()
- virtual const char ∗ GetWindowName ()
- void PrintSelf (ostream &os, vtkIndent indent)
- virtual void Render (void)
- virtual void SetColorLevel (double s)
- virtual void SetColorWindow (double s)
- virtual void SetDisplayId (void ∗a)
- virtual void SetInput (vtkImageData ∗in)
- virtual void SetInputConnection (vtkAlgorithmOutput ∗input)
- virtual void SetOffScreenRendering (int)
- void SetOverlayVisibility (double vis)
- virtual void SetParentId (void ∗a)
- virtual void SetPosition (int a, int b)
- virtual void SetPosition (int a[2])
- virtual void SetRenderer (vtkRenderer ∗arg)
- virtual void SetRenderWindow (vtkRenderWindow ∗arg)
- virtual void SetSize (int a, int b)
- virtual void SetSize (int a[2])
- virtual void SetSlice (int s)
- virtual void SetSliceOrientation (int orientation)
- virtual void SetSliceOrientationToXY ()
- virtual void SetSliceOrientationToXZ ()
- virtual void SetSliceOrientationToYZ ()
- virtual void SetupInteractor (vtkRenderWindowInteractor ∗)
- virtual void SetWindowId (void ∗a)
- virtual void UpdateDisplayExtent ()
- VTK_LEGACY (int GetWholeZMin())
- VTK_LEGACY (int GetWholeZMax())
- VTK_LEGACY (int GetZSlice())
- VTK_LEGACY (void SetZSlice(int))
- vtkBooleanMacro (OffScreenRendering, int)
- vtkGetMacro (SliceOrientation, int)
- vtkGetMacro (Slice, int)
- vtkGetObjectMacro (RenderWindow, vtkRenderWindow)
- vtkGetObjectMacro (Renderer, vtkRenderer)
- vtkGetObjectMacro (ImageActor, vtkImageActor)
- vtkGetObjectMacro (WindowLevel, vtkImageMapToWindowLevelColors2)
- vtkGetObjectMacro (InteractorStyle, vtkInteractorStyleImage)
- vtkTypeRevisionMacro (vtkImageColorViewer, vtkObject)

Static Public Member Functions

- static vtkImageColorViewer ∗ New ()

Protected Member Functions

- vtkImageColorViewer ()
- ∼vtkImageColorViewer ()
- virtual void InstallPipeline ()
- virtual void UnInstallPipeline ()
- virtual void UpdateOrientation ()

Protected Attributes

- int FirstRender
- vtkImageActor ∗ ImageActor
- vtkRenderWindowInteractor ∗ Interactor
- vtkInteractorStyleImage ∗ InteractorStyle
- vtkImageActor ∗ OverlayImageActor
- vtkRenderer ∗ Renderer
- vtkRenderWindow ∗ RenderWindow
- int Slice
- int SliceOrientation
- vtkImageMapToWindowLevelColors2 ∗ WindowLevel

25.330.1 Detailed Description

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

25.330.2 Member Enumeration Documentation

25.330.2.1 anonymous enum

Enumerator:

SLICE_ORIENTATION_YZ

SLICE_ORIENTATION_XZ

SLICE_ORIENTATION_XY

25.330.3 Constructor & Destructor Documentation

25.330.3.1 vtkImageColorViewer::vtkImageColorViewer ( ) [protected]

25.330.3.2 vtkImageColorViewer::∼vtkImageColorViewer ( ) [protected]

25.330.4 Member Function Documentation

25.330.4.1 virtual void vtkImageColorViewer::AddInput (vtkImageData ∗ *input*) [virtual]

25.330.4.2 virtual void vtkImageColorViewer::AddInputConnection (vtkAlgorithmOutput ∗ *input*) [virtual]

25.330.4.3 virtual double vtkImageColorViewer::GetColorLevel ( ) [virtual]

25.330.4.4 virtual double vtkImageColorViewer::GetColorWindow ( ) [virtual]

25.330.4.5 virtual vtkImageData* vtkImageColorViewer::GetInput ( ) [virtual]

25.330.4.6 virtual int vtkImageColorViewer::GetOffScreenRendering ( ) [virtual]

25.330.4.7 double vtkImageColorViewer::GetOverlayVisibility ( )

25.330.4.8 virtual int* vtkImageColorViewer::GetPosition ( ) [virtual]

25.330.4.9 virtual int* vtkImageColorViewer::GetSize ( ) [virtual]

25.330.4.10 virtual int vtkImageColorViewer::GetSliceMax ( ) [virtual]

25.330.4.11 virtual int vtkImageColorViewer::GetSliceMin ( ) [virtual]

25.330.4.12 virtual void vtkImageColorViewer::GetSliceRange (int *range[2]*) [inline], [virtual]

References GetSliceRange().

Referenced by GetSliceRange().

25.330.4.13 virtual void vtkImageColorViewer::GetSliceRange (int & *min*, int & *max*) [virtual]

25.330.4.14 virtual int* vtkImageColorViewer::GetSliceRange ( ) [virtual]

25.330.4.15 virtual const char* vtkImageColorViewer::GetWindowName ( ) [virtual]

25.330.4.16 virtual void vtkImageColorViewer::InstallPipeline ( ) [protected], [virtual]

25.330.4.17 static vtkImageColorViewer* vtkImageColorViewer::New ( ) [static]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

25.330.4.18 void vtkImageColorViewer::PrintSelf (ostream & *os*, vtkIndent *indent*)

25.330.4.19 virtual void vtkImageColorViewer::Render (void) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

25.330.4.20 virtual void vtkImageColorViewer::SetColorLevel (double *s*) [virtual]

25.330.4.21 virtual void vtkImageColorViewer::SetColorWindow (double *s*) [virtual]

25.330.4.22 virtual void vtkImageColorViewer::SetDisplayId (void * *a*) [virtual]

25.330.4.23 virtual void vtkImageColorViewer::SetInput (vtkImageData * *in*) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

25.330.4.24 virtual void vtkImageColorViewer::SetInputConnection (vtkAlgorithmOutput * *input*) [virtual]

25.330.4.25 virtual void vtkImageColorViewer::SetOffScreenRendering (int) [virtual]

25.330.4.26 void vtkImageColorViewer::SetOverlayVisibility (double *vis*)

25.330.4.27 virtual void vtkImageColorViewer::SetParentId (void * *a*) [virtual]

25.330.4.28 virtual void vtkImageColorViewer::SetPosition (int *a*, int *b*) [virtual]

25.330.4.29 virtual void vtkImageColorViewer::SetPosition (int *a[2]*) [inline], [virtual]

References SetPosition().

Referenced by SetPosition().

25.330.4.30 virtual void vtkImageColorViewer::SetRenderer (vtkRenderer * *arg*) [virtual]

25.330.4.31 virtual void vtkImageColorViewer::SetRenderWindow (vtkRenderWindow * *arg*) [virtual]

25.330.4.32 virtual void vtkImageColorViewer::SetSize (int *a*, int *b*) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

25.330.4.33 virtual void vtkImageColorViewer::SetSize (int *a[2]*) [inline], [virtual]

References SetSize().

Referenced by SetSize().

25.330.4.34 virtual void vtkImageColorViewer::SetSlice (int *s*) [virtual]

25.330.4.35 virtual void vtkImageColorViewer::SetSliceOrientation (int *orientation*) [virtual]

25.330.4.36 virtual void vtkImageColorViewer::SetSliceOrientationToXY ( ) [inline], [virtual]

References SLICE_ORIENTATION_XY.

25.330.4.37 virtual void vtkImageColorViewer::SetSliceOrientationToXZ ( ) [inline], [virtual]

References SLICE_ORIENTATION_XZ.

25.330.4.38 virtual void vtkImageColorViewer::SetSliceOrientationToYZ ( ) [inline],[virtual]

References SLICE_ORIENTATION_YZ.

25.330.4.39 virtual void vtkImageColorViewer::SetupInteractor (vtkRenderWindowInteractor *) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

25.330.4.40 virtual void vtkImageColorViewer::SetWindowId (void * *a*) [virtual]

25.330.4.41 virtual void vtkImageColorViewer::UnInstallPipeline ( ) [protected],[virtual]

25.330.4.42 virtual void vtkImageColorViewer::UpdateDisplayExtent ( ) [virtual]

25.330.4.43 virtual void vtkImageColorViewer::UpdateOrientation ( ) [protected],[virtual]

25.330.4.44 vtkImageColorViewer::VTK_LEGACY (int *GetWholeZMin()*)

25.330.4.45 vtkImageColorViewer::VTK_LEGACY (int *GetWholeZMax()*)

25.330.4.46 vtkImageColorViewer::VTK_LEGACY (int *GetZSlice()*)

25.330.4.47 vtkImageColorViewer::VTK_LEGACY (void *SetZSliceint*)

25.330.4.48 vtkImageColorViewer::vtkBooleanMacro (OffScreenRendering , int)

25.330.4.49 vtkImageColorViewer::vtkGetMacro (**SliceOrientation** , int)

25.330.4.50 vtkImageColorViewer::vtkGetMacro (**Slice** , int)

25.330.4.51 vtkImageColorViewer::vtkGetObjectMacro (**RenderWindow** , vtkRenderWindow)

25.330.4.52 vtkImageColorViewer::vtkGetObjectMacro (**Renderer** , vtkRenderer)

25.330.4.53 vtkImageColorViewer::vtkGetObjectMacro (**ImageActor** , vtkImageActor)

25.330.4.54 vtkImageColorViewer::vtkGetObjectMacro (**WindowLevel** , **vtkImageMapToWindowLevelColors2**)

25.330.4.55 vtkImageColorViewer::vtkGetObjectMacro (**InteractorStyle** , vtkInteractorStyleImage)

25.330.4.56 vtkImageColorViewer::vtkTypeRevisionMacro (**vtkImageColorViewer** , vtkObject)

25.330.5 Member Data Documentation

25.330.5.1 int vtkImageColorViewer::FirstRender [protected]

25.330.5.2 vtkImageActor* vtkImageColorViewer::ImageActor [protected]

25.330.5.3 vtkRenderWindowInteractor* vtkImageColorViewer::Interactor [protected]

25.330.5.4 vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle [protected]

25.330.5.5 vtkImageActor* vtkImageColorViewer::OverlayImageActor [protected]

25.330.5.6 vtkRenderer* vtkImageColorViewer::Renderer [protected]

25.330.5.7 vtkRenderWindow* vtkImageColorViewer::RenderWindow [protected]

25.330.5.8 int vtkImageColorViewer::Slice [protected]

25.330.5.9 int vtkImageColorViewer::SliceOrientation [protected]

25.330.5.10 vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel [protected]

The documentation for this class was generated from the following file:

- vtkImageColorViewer.h

25.331 vtkImageMapToColors16 Class Reference

```
#include <vtkImageMapToColors16.h>
```

Public Member Functions

- virtual unsigned long GetMTime ()
- void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetLookupTable (vtkScalarsToColors *)
- void SetOutputFormatToLuminance ()
- void SetOutputFormatToLuminanceAlpha ()
- void SetOutputFormatToRGB ()
- void SetOutputFormatToRGBA ()
- vtkBooleanMacro (PassAlphaToOutput, int)
- vtkGetMacro (OutputFormat, int)
- vtkGetMacro (ActiveComponent, int)
- vtkGetMacro (PassAlphaToOutput, int)
- vtkGetObjectMacro (LookupTable, vtkScalarsToColors)
- vtkSetMacro (OutputFormat, int)
- vtkSetMacro (ActiveComponent, int)
- vtkSetMacro (PassAlphaToOutput, int)
- vtkTypeRevisionMacro (vtkImageMapToColors16, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageMapToColors16 * New ()

Protected Member Functions

- vtkImageMapToColors16 ()
- ~vtkImageMapToColors16 ()
- virtual int RequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- void ThreadedRequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int extent[6], int id)

Protected Attributes

- int ActiveComponent
- int DataWasPassed
- vtkScalarsToColors * LookupTable
- int OutputFormat
- int PassAlphaToOutput

25.331.1 Constructor & Destructor Documentation

25.331.1.1 vtkImageMapToColors16::vtkImageMapToColors16 ( ) [protected]

25.331.1.2 vtkImageMapToColors16::~vtkImageMapToColors16 ( ) [protected]

25.331.2 Member Function Documentation

25.331.2.1 virtual unsigned long vtkImageMapToColors16::GetMTime ( ) [virtual]

25.331.2.2 static vtkImageMapToColors16* vtkImageMapToColors16::New ( ) [static]

25.331.2.3 void vtkImageMapToColors16::PrintSelf (ostream & *os*, vtkIndent *indent*)

25.331.2.4 virtual int vtkImageMapToColors16::RequestData (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*) [protected], [virtual]

25.331.2.5 virtual int vtkImageMapToColors16::RequestInformation (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected], [virtual]

25.331.2.6 virtual void vtkImageMapToColors16::SetLookupTable (vtkScalarsToColors *) [virtual]

25.331.2.7 void vtkImageMapToColors16::SetOutputFormatToLuminance ( ) [inline]

25.331.2.8 void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ( ) [inline]

25.331.2.9 void vtkImageMapToColors16::SetOutputFormatToRGB ( ) [inline]

25.331.2.10 void vtkImageMapToColors16::SetOutputFormatToRGBA ( ) [inline]

25.331.2.11 void vtkImageMapToColors16::ThreadedRequestData (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*, vtkImageData *** *inData*, vtkImageData ** *outData*, int *extent[6]*, int *id*) [protected]

25.331.2.12 vtkImageMapToColors16::vtkBooleanMacro (PassAlphaToOutput , int)

25.331.2.13 vtkImageMapToColors16::vtkGetMacro (OutputFormat , int)

25.331.2.14 vtkImageMapToColors16::vtkGetMacro (ActiveComponent , int)

25.331.2.15 vtkImageMapToColors16::vtkGetMacro (PassAlphaToOutput , int)

25.331.2.16 vtkImageMapToColors16::vtkGetObjectMacro (LookupTable , vtkScalarsToColors)

25.331.2.17 vtkImageMapToColors16::vtkSetMacro (OutputFormat , int)

25.331.2.18 vtkImageMapToColors16::vtkSetMacro (ActiveComponent , int)

25.331.2.19 vtkImageMapToColors16::vtkSetMacro (PassAlphaToOutput , int)

25.331.2.20 vtkImageMapToColors16::vtkTypeRevisionMacro (vtkImageMapToColors16 , vtkThreadedImageAlgorithm)

25.331.3 Member Data Documentation

25.331.3.1 int vtkImageMapToColors16::ActiveComponent [protected]

25.331.3.2 int vtkImageMapToColors16::DataWasPassed [protected]

25.331.3.3 vtkScalarsToColors* vtkImageMapToColors16::LookupTable [protected]

25.331.3.4 int vtkImageMapToColors16::OutputFormat [protected]

25.331.3.5 int vtkImageMapToColors16::PassAlphaToOutput [protected]

The documentation for this class was generated from the following file:

- vtkImageMapToColors16.h

25.332 vtkImageMapToWindowLevelColors2 Class Reference

```
#include <vtkImageMapToWindowLevelColors2.h>
```

Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetMacro (Window, double)
- vtkGetMacro (Level, double)
- vtkSetMacro (Window, double)
- vtkSetMacro (Level, double)
- vtkTypeRevisionMacro (vtkImageMapToWindowLevelColors2, vtkImageMapToColors)

Static Public Member Functions

- static vtkImageMapToWindowLevelColors2 ∗ New ()

Protected Member Functions

- vtkImageMapToWindowLevelColors2 ()
- ∼vtkImageMapToWindowLevelColors2 ()
- virtual int RequestData (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗outputVector)
- virtual int RequestInformation (vtkInformation ∗, vtkInformationVector ∗∗, vtkInformationVector ∗)
- void ThreadedRequestData (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗outputVector, vtkImageData ∗∗∗inData, vtkImageData ∗∗outData, int extent[6], int id)

Protected Attributes

- double Level
- double Window

25.332.1 Constructor & Destructor Documentation

25.332.1.1 vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ( ) [protected]

25.332.1.2 vtkImageMapToWindowLevelColors2::∼vtkImageMapToWindowLevelColors2 ( ) [protected]

25.332.2 Member Function Documentation

25.332.2.1 static vtkImageMapToWindowLevelColors2∗ vtkImageMapToWindowLevelColors2::New ( ) [static]

25.332.2.2 void vtkImageMapToWindowLevelColors2::PrintSelf (ostream & *os,* vtkIndent *indent*)

25.332.2.3 virtual int vtkImageMapToWindowLevelColors2::RequestData (vtkInformation ∗ *request,* vtkInformationVector ∗∗ *inputVector,* vtkInformationVector ∗ *outputVector*) [protected], [virtual]

25.332.2.4 virtual int vtkImageMapToWindowLevelColors2::RequestInformation (vtkInformation ∗ , vtkInformationVector ∗∗ , vtkInformationVector ∗) [protected], [virtual]

25.332.2.5 void vtkImageMapToWindowLevelColors2::ThreadedRequestData (vtkInformation ∗ *request,* vtkInformationVector ∗∗ *inputVector,* vtkInformationVector ∗ *outputVector,* vtkImageData ∗∗∗ *inData,* vtkImageData ∗∗ *outData,* int *extent[6],* int *id*) [protected]

25.332.2.6 vtkImageMapToWindowLevelColors2::vtkGetMacro (Window , double)

25.332.2.7 vtkImageMapToWindowLevelColors2::vtkGetMacro (Level , double)

25.332.2.8 vtkImageMapToWindowLevelColors2::vtkSetMacro (Window , double)

25.332.2.9 vtkImageMapToWindowLevelColors2::vtkSetMacro (Level , double)

25.332.2.10 vtkImageMapToWindowLevelColors2::vtkTypeRevisionMacro (vtkImageMapToWindowLevelColors2 , vtkImageMapToColors)

25.332.3 Member Data Documentation

25.332.3.1 double vtkImageMapToWindowLevelColors2::Level [protected]

25.332.3.2 double vtkImageMapToWindowLevelColors2::Window [protected]

The documentation for this class was generated from the following file:

- vtkImageMapToWindowLevelColors2.h

25.333 vtkImagePlanarComponentsToComponents Class Reference

```
#include <vtkImagePlanarComponentsToComponents.h>
```

Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (vtkImagePlanarComponentsToComponents, vtkImageAlgorithm)

Static Public Member Functions

- static vtkImagePlanarComponentsToComponents ∗ New ()

Protected Member Functions

- vtkImagePlanarComponentsToComponents ()
- ∼vtkImagePlanarComponentsToComponents ()
- virtual int RequestData (vtkInformation ∗, vtkInformationVector ∗∗, vtkInformationVector ∗)

25.333.1 Constructor & Destructor Documentation

25.333.1.1 vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ( ) [protected]

25.333.1.2 vtkImagePlanarComponentsToComponents::∼vtkImagePlanarComponentsToComponents ( ) [inline], [protected]

25.333.2 Member Function Documentation

25.333.2.1 static vtkImagePlanarComponentsToComponents∗ vtkImagePlanarComponentsToComponents::New ( ) [static]

25.333.2.2 void vtkImagePlanarComponentsToComponents::PrintSelf (ostream & *os,* vtkIndent *indent*)

25.333.2.3 virtual int vtkImagePlanarComponentsToComponents::RequestData (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected],[virtual]

25.333.2.4 vtkImagePlanarComponentsToComponents::vtkTypeRevisionMacro (vtkImagePlanarComponentsToComponents , vtkImageAlgorithm)

The documentation for this class was generated from the following file:

- vtkImagePlanarComponentsToComponents.h

25.334 vtkImageRGBToYBR Class Reference

```
#include <vtkImageRGBToYBR.h>
```

Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (vtkImageRGBToYBR, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageRGBToYBR * New ()

Protected Member Functions

- vtkImageRGBToYBR ()
- ~vtkImageRGBToYBR ()
- void ThreadedExecute (vtkImageData *inData, vtkImageData *outData, int ext[6], int id)

25.334.1 Constructor & Destructor Documentation

25.334.1.1 vtkImageRGBToYBR::vtkImageRGBToYBR ( ) [protected]

25.334.1.2 vtkImageRGBToYBR::~vtkImageRGBToYBR ( ) [inline],[protected]

25.334.2 Member Function Documentation

25.334.2.1 static vtkImageRGBToYBR* vtkImageRGBToYBR::New ( ) [static]

25.334.2.2 void vtkImageRGBToYBR::PrintSelf (ostream & *os,* vtkIndent *indent*)

25.334.2.3 void vtkImageRGBToYBR::ThreadedExecute (vtkImageData * *inData,* vtkImageData * *outData,* int *ext[6],* int *id*) [protected]

25.334.2.4 vtkImageRGBToYBR::vtkTypeRevisionMacro (vtkImageRGBToYBR , vtkThreadedImageAlgorithm)

The documentation for this class was generated from the following file:

- vtkImageRGBToYBR.h

25.335 vtkImageYBRToRGB Class Reference

```
#include <vtkImageYBRToRGB.h>
```

Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (vtkImageYBRToRGB, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageYBRToRGB ∗ New ()

Protected Member Functions

- vtkImageYBRToRGB ()
- ∼vtkImageYBRToRGB ()
- void ThreadedExecute (vtkImageData ∗inData, vtkImageData ∗outData, int ext[6], int id)

25.335.1 Constructor & Destructor Documentation

25.335.1.1 vtkImageYBRToRGB::vtkImageYBRToRGB ( ) [protected]

25.335.1.2 vtkImageYBRToRGB::∼vtkImageYBRToRGB ( ) [inline],[protected]

25.335.2 Member Function Documentation

25.335.2.1 static vtkImageYBRToRGB∗ vtkImageYBRToRGB::New ( ) [static]

25.335.2.2 void vtkImageYBRToRGB::PrintSelf (ostream & *os,* vtkIndent *indent*)

25.335.2.3 void vtkImageYBRToRGB::ThreadedExecute (vtkImageData ∗ *inData,* vtkImageData ∗ *outData,* int *ext[6],* int *id*) [protected]

25.335.2.4 vtkImageYBRToRGB::vtkTypeRevisionMacro (vtkImageYBRToRGB , vtkThreadedImageAlgorithm)

The documentation for this class was generated from the following file:

- vtkImageYBRToRGB.h

25.336 vtkLookupTable16 Class Reference

```
#include <vtkLookupTable16.h>
```

Public Member Functions

- void Build ()
- unsigned short ∗ GetPointer (const vtkIdType id)
- void PrintSelf (ostream &os, vtkIndent indent)
- void SetNumberOfTableValues (vtkIdType number)
- vtkTypeRevisionMacro (vtkLookupTable16, vtkLookupTable)
- unsigned char ∗ WritePointer (const vtkIdType id, const int number)

Static Public Member Functions

- static vtkLookupTable16 ∗ New ()

Protected Member Functions

- vtkLookupTable16 (int sze=256, int ext=256)
- ∼vtkLookupTable16 ()
- void MapScalarsThroughTable2 (void ∗input, unsigned char ∗output, int inputDataType, int numberOfValues, int inputIncrement, int outputFormat)

Protected Attributes

- vtkUnsignedShortArray ∗ Table16

25.336.1 Constructor & Destructor Documentation

25.336.1.1 vtkLookupTable16::vtkLookupTable16 (int *sze* = 256, int *ext* = 256) [protected]

25.336.1.2 vtkLookupTable16::∼vtkLookupTable16 ( ) [protected]

25.336.2 Member Function Documentation

25.336.2.1 void vtkLookupTable16::Build ( )

25.336.2.2 unsigned short∗ vtkLookupTable16::GetPointer (const vtkIdType *id*) [inline]

25.336.2.3 void vtkLookupTable16::MapScalarsThroughTable2 (void ∗ *input*, unsigned char ∗ *output*, int *inputDataType*, int *numberOfValues*, int *inputIncrement*, int *outputFormat*) [protected]

25.336.2.4 static vtkLookupTable16∗ vtkLookupTable16::New ( ) [static]

25.336.2.5 void vtkLookupTable16::PrintSelf (ostream & *os*, vtkIndent *indent*)

25.336.2.6 void vtkLookupTable16::SetNumberOfTableValues (vtkIdType *number*)

25.336.2.7 vtkLookupTable16::vtkTypeRevisionMacro (vtkLookupTable16 , vtkLookupTable)

25.336.2.8 unsigned char ∗ vtkLookupTable16::WritePointer (const vtkIdType *id*, const int *number*) [inline]

References Table16.

25.336.3 Member Data Documentation

25.336.3.1 vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]

Referenced by WritePointer().

The documentation for this class was generated from the following file:

- vtkLookupTable16.h

25.337 vtkRTStructSetProperties Class Reference

```
#include <vtkRTStructSetProperties.h>
```

Public Member Functions

- void AddContourReferencedFrameOfReference (vtkIdType pdnum, const char *classuid, const char *instanceuid)
- void AddReferencedFrameOfReference (const char *classuid, const char *instanceuid)
- void AddStructureSetROI (int roinumber, const char *reframerefuid, const char *roiname, const char *ROI-GenerationAlgorithm)
- void AddStructureSetROIObservation (int refnumber, int observationnumber, const char *rtroiinterpretedtype, const char *roiinterpreter)
- virtual void Clear ()
- virtual void DeepCopy (vtkRTStructSetProperties *p)
- const char * GetContourReferencedFrameOfReferenceClassUID (vtkIdType pdnum, vtkIdType id)
- const char * GetContourReferencedFrameOfReferenceInstanceUID (vtkIdType pdnum, vtkIdType id)
- vtkIdType GetNumberOfContourReferencedFrameOfReferences ()
- vtkIdType GetNumberOfContourReferencedFrameOfReferences (vtkIdType pdnum)
- vtkIdType GetNumberOfReferencedFrameOfReferences ()
- vtkIdType GetNumberOfStructureSetROIs ()
- const char * GetReferencedFrameOfReferenceClassUID (vtkIdType id)
- const char * GetReferencedFrameOfReferenceInstanceUID (vtkIdType id)
- int GetStructureSetObservationNumber (vtkIdType id)
- const char * GetStructureSetROIGenerationAlgorithm (vtkIdType)
- const char * GetStructureSetROIName (vtkIdType)
- int GetStructureSetROINumber (vtkIdType id)
- const char * GetStructureSetROIRefFrameRefUID (vtkIdType)
- const char * GetStructureSetRTROIInterpretedType (vtkIdType id)
- void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetStringMacro (StructureSetLabel)
- vtkGetStringMacro (StructureSetName)
- vtkGetStringMacro (StructureSetDate)
- vtkGetStringMacro (StructureSetTime)
- vtkGetStringMacro (SOPInstanceUID)
- vtkGetStringMacro (StudyInstanceUID)
- vtkGetStringMacro (SeriesInstanceUID)
- vtkGetStringMacro (ReferenceSeriesInstanceUID)
- vtkGetStringMacro (ReferenceFrameOfReferenceUID)
- vtkSetStringMacro (StructureSetLabel)
- vtkSetStringMacro (StructureSetName)

- vtkSetStringMacro (StructureSetDate)
- vtkSetStringMacro (StructureSetTime)
- vtkSetStringMacro (SOPInstanceUID)
- vtkSetStringMacro (StudyInstanceUID)
- vtkSetStringMacro (SeriesInstanceUID)
- vtkSetStringMacro (ReferenceSeriesInstanceUID)
- vtkSetStringMacro (ReferenceFrameOfReferenceUID)
- vtkTypeRevisionMacro (vtkRTStructSetProperties, vtkObject)

Static Public Member Functions

- static vtkRTStructSetProperties ∗ New ()

Protected Member Functions

- vtkRTStructSetProperties ()
- ∼vtkRTStructSetProperties ()

Protected Attributes

- vtkRTStructSetPropertiesInternals ∗ Internals
- char ∗ ReferenceFrameOfReferenceUID
- char ∗ ReferenceSeriesInstanceUID
- char ∗ SeriesInstanceUID
- char ∗ SOPInstanceUID
- char ∗ StructureSetDate
- char ∗ StructureSetLabel
- char ∗ StructureSetName
- char ∗ StructureSetTime
- char ∗ StudyInstanceUID

25.337.1 Detailed Description

Examples:

GenerateRTSTRUCT.cxx.

25.337.2 Constructor & Destructor Documentation

25.337.2.1 vtkRTStructSetProperties::vtkRTStructSetProperties ( ) [protected]

25.337.2.2 vtkRTStructSetProperties::∼vtkRTStructSetProperties ( ) [protected]

25.337.3 Member Function Documentation

25.337.3.1 void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (vtkIdType *pdnum*, const char ∗ *classuid*, const char ∗ *instanceuid*)

25.337.3.2 void vtkRTStructSetProperties::AddReferencedFrameOfReference (const char ∗ *classuid*, const char ∗ *instanceuid*)

25.337.3.3 void vtkRTStructSetProperties::AddStructureSetROI (int *roinumber,* const char * *reframerefuid,* const char * *roiname,* const char * *ROIGenerationAlgorithm*)

25.337.3.4 void vtkRTStructSetProperties::AddStructureSetROIObservation (int *refnumber,* int *observationnumber,* const char * *rtroiinterpretedtype,* const char * *roiinterpreter*)

25.337.3.5 virtual void vtkRTStructSetProperties::Clear ( ) [virtual]

25.337.3.6 virtual void vtkRTStructSetProperties::DeepCopy (**vtkRTStructSetProperties** * *p*) [virtual]

25.337.3.7 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (vtkIdType *pdnum,* vtkIdType *id*)

25.337.3.8 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (vtkIdType *pdnum,* vtkIdType *id*)

25.337.3.9 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( )

25.337.3.10 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (vtkIdType *pdnum*)

25.337.3.11 vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ( )

25.337.3.12 vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ( )

25.337.3.13 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (vtkIdType *id*)

25.337.3.14 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (vtkIdType *id*)

25.337.3.15 int vtkRTStructSetProperties::GetStructureSetObservationNumber (vtkIdType *id*)

25.337.3.16 const char* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (vtkIdType)

25.337.3.17 const char* vtkRTStructSetProperties::GetStructureSetROIName (vtkIdType)

25.337.3.18 int vtkRTStructSetProperties::GetStructureSetROINumber (vtkIdType *id*)

25.337.3.19 const char* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (vtkIdType)

25.337.3.20 const char* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (vtkIdType *id*)

25.337.3.21 static **vtkRTStructSetProperties*** vtkRTStructSetProperties::New ( ) [static]

Examples:

GenerateRTSTRUCT.cxx.

25.337.3.22 void vtkRTStructSetProperties::PrintSelf (ostream & *os,* vtkIndent *indent*)

25.337.3.23 vtkRTStructSetProperties::vtkGetStringMacro (**StructureSetLabel**)

25.337.3.24 vtkRTStructSetProperties::vtkGetStringMacro (**StructureSetName**)

25.337.3.25 vtkRTStructSetProperties::vtkGetStringMacro (StructureSetDate)

25.337.3.26 vtkRTStructSetProperties::vtkGetStringMacro (StructureSetTime)

25.337.3.27 vtkRTStructSetProperties::vtkGetStringMacro (SOPInstanceUID)

25.337.3.28 vtkRTStructSetProperties::vtkGetStringMacro (StudyInstanceUID)

25.337.3.29 vtkRTStructSetProperties::vtkGetStringMacro (SeriesInstanceUID)

25.337.3.30 vtkRTStructSetProperties::vtkGetStringMacro (ReferenceSeriesInstanceUID)

25.337.3.31 vtkRTStructSetProperties::vtkGetStringMacro (ReferenceFrameOfReferenceUID)

25.337.3.32 vtkRTStructSetProperties::vtkSetStringMacro (StructureSetLabel)

25.337.3.33 vtkRTStructSetProperties::vtkSetStringMacro (StructureSetName)

25.337.3.34 vtkRTStructSetProperties::vtkSetStringMacro (StructureSetDate)

25.337.3.35 vtkRTStructSetProperties::vtkSetStringMacro (StructureSetTime)

25.337.3.36 vtkRTStructSetProperties::vtkSetStringMacro (SOPInstanceUID)

25.337.3.37 vtkRTStructSetProperties::vtkSetStringMacro (StudyInstanceUID)

25.337.3.38 vtkRTStructSetProperties::vtkSetStringMacro (SeriesInstanceUID)

25.337.3.39 vtkRTStructSetProperties::vtkSetStringMacro (ReferenceSeriesInstanceUID)

25.337.3.40 vtkRTStructSetProperties::vtkSetStringMacro (ReferenceFrameOfReferenceUID)

25.337.3.41 vtkRTStructSetProperties::vtkTypeRevisionMacro (vtkRTStructSetProperties , vtkObject)

25.337.4 Member Data Documentation

25.337.4.1 vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals [protected]

25.337.4.2 char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID [protected]

25.337.4.3 char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID [protected]

25.337.4.4 char* vtkRTStructSetProperties::SeriesInstanceUID [protected]

25.337.4.5 char* vtkRTStructSetProperties::SOPInstanceUID [protected]

25.337.4.6 char* vtkRTStructSetProperties::StructureSetDate [protected]

25.337.4.7 char* vtkRTStructSetProperties::StructureSetLabel [protected]

25.337.4.8 char* vtkRTStructSetProperties::StructureSetName [protected]

25.337.4.9 char* vtkRTStructSetProperties::StructureSetTime [protected]

25.337.4.10 char* vtkRTStructSetProperties::StudyInstanceUID [protected]

The documentation for this class was generated from the following file:

- vtkRTStructSetProperties.h

25.338 gdcm::Waveform Class Reference

Waveform class.

```
#include <gdcmWaveform.h>
```

Public Member Functions

- Waveform ()

25.338.1 Detailed Description

Waveform class.

25.338.2 Constructor & Destructor Documentation

25.338.2.1 gdcm::Waveform::Waveform ( ) [inline]

The documentation for this class was generated from the following file:

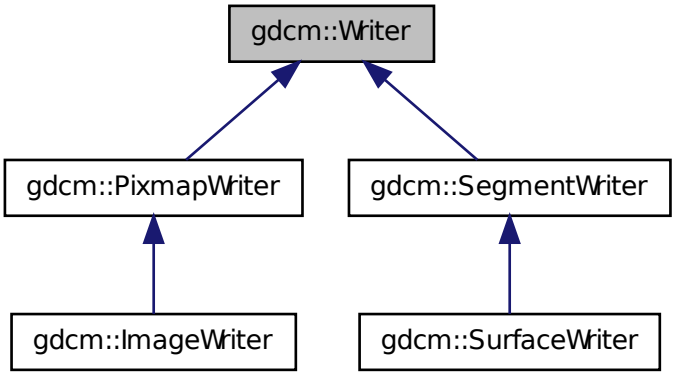
- gdcmWaveform.h

25.339 gdcm::Writer Class Reference

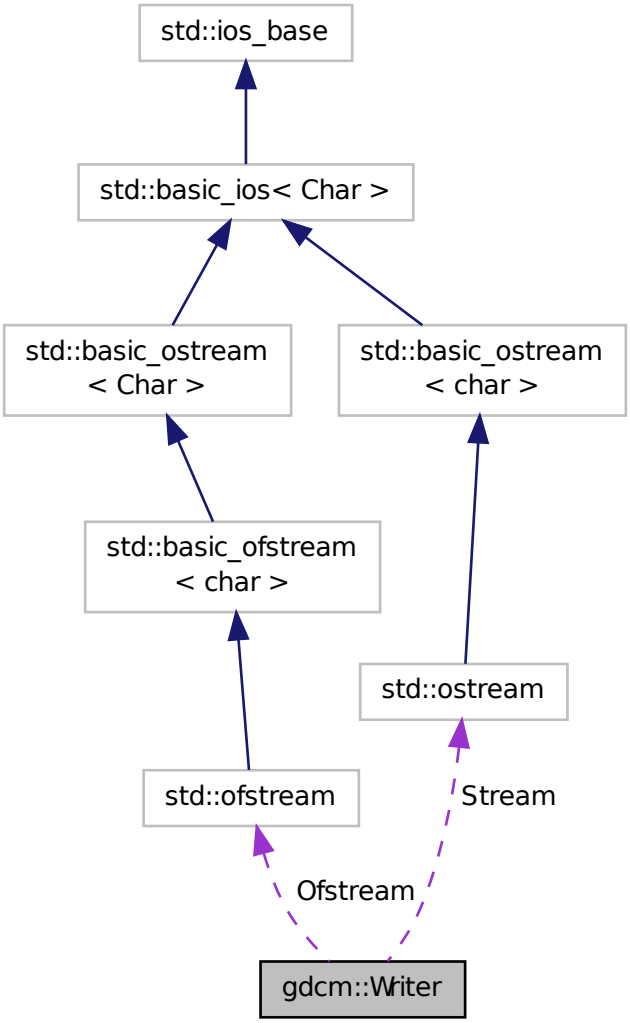
Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.

```
#include <gdcmWriter.h>
```

Inheritance diagram for gdcm::Writer:



Collaboration diagram for gdcm::Writer:



Public Member Functions

- Writer ()
- virtual ~Writer ()
- void CheckFileMetaInformationOff ()
- void CheckFileMetaInformationOn ()
- File & GetFile ()
- void SetCheckFileMetaInformation (bool b)

  Undocumented function, do not use (= leave default)
- void SetFile (const File &f)

Set/Get the DICOM file (DataSet + Header)

- void SetFileName (const char *filename_native)

  Set the filename of DICOM file to write:

- void SetStream (std::ostream &output_stream)

  Set user ostream buffer.

- virtual bool Write ()

  Main function to tell the writer to write.

Protected Member Functions

- std::ostream * GetStreamPtr () const
- void SetWriteDataSetOnly (bool b)

Protected Attributes

- std::ofstream * Ofstream
- std::ostream * Stream

Friends

- class StreamImageWriter

25.339.1 Detailed Description

Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.

Detailled description here To avoid GDCM being yet another broken DICOM lib we try to be user level and avoid writing illegal stuff (odd length, non-zero value for Item start/end length ...) Therefore you cannot (well unless you are really smart) write DICOM with even length tag. All the checks are consider basics:

- Correct Meta Information Header (see gdcm::FileMetaInformation)
- Zero value for Item Length (0xfffe, 0xe00d/0xe0dd)
- Even length for any elements
- Alphabetical order for elements (garanteed by design of internals)
- 32bits VR will be rewritten with 00

Warning

gdcm::Writer cannot write a DataSet if no SOP Instance UID (0008,0018) is found, unless a DICOMDIR is being written out

See also

Reader DataSet File

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

25.339.2 Constructor & Destructor Documentation

25.339.2.1 gdcm::Writer::Writer ( )

25.339.2.2 virtual gdcm::Writer::~Writer ( ) [virtual]

25.339.3 Member Function Documentation

25.339.3.1 void gdcm::Writer::CheckFileMetaInformationOff ( ) [inline]

Examples:

FixBrokenJ2K.cxx, and HelloWorld.cxx.

25.339.3.2 void gdcm::Writer::CheckFileMetaInformationOn ( ) [inline]

25.339.3.3 File& gdcm::Writer::GetFile ( ) [inline]

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

25.339.3.4 std::ostream* gdcm::Writer::GetStreamPtr ( ) const [inline], [protected]

25.339.3.5 void gdcm::Writer::SetCheckFileMetaInformation (bool *b*) [inline]

Undocumented function, do not use (= leave default)

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, and PatchFile.cxx.

25.339.3.6 void gdcm::Writer::SetFile (const File & *f*) [inline]

Set/Get the DICOM file (DataSet + Header)

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, DuplicatePCDE.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

25.339.3.7 void gdcm::Writer::SetFileName (const char * *filename_native*)

Set the filename of DICOM file to write:

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloVizWorld.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

25.339.3.8 void gdcm::Writer::SetStream (std::ostream & *output_stream*) [inline]

Set user ostream buffer.

25.339.3.9 void gdcm::Writer::SetWriteDataSetOnly (bool *b*) [inline], [protected]

25.339.3.10 virtual bool gdcm::Writer::Write ( ) [virtual]

Main function to tell the writer to write.

Reimplemented in gdcm::PixmapWriter, gdcm::ImageWriter, gdcm::SurfaceWriter, and gdcm::SegmentWriter.

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

25.339.4 Friends And Related Function Documentation

25.339.4.1 friend class StreamImageWriter [friend]

25.339.5 Member Data Documentation

25.339.5.1 std::ofstream* gdcm::Writer::Ofstream [protected]

25.339.5.2 std::ostream* gdcm::Writer::Stream [protected]

The documentation for this class was generated from the following file:

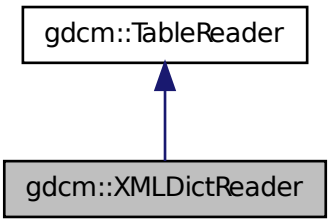
- gdcmWriter.h

25.340 gdcm::XMLDictReader Class Reference

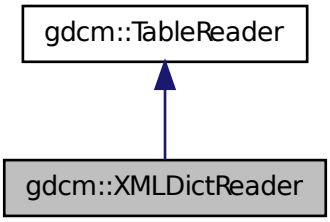
Class for representing a XMLDictReader.

```
#include <gdcmXMLDictReader.h>
```

Inheritance diagram for gdcm::XMLDictReader:



Collaboration diagram for gdcm::XMLDictReader:



Public Member Functions

- XMLDictReader ()
- ~XMLDictReader ()
- void CharacterDataHandler (const char *data, int length)
- void EndElement (const char *name)
- const Dict & GetDict ()
- void StartElement (const char *name, const char **atts)

Protected Member Functions

- void HandleDescription (const char **atts)

- void HandleEntry (const char **atts)

25.340.1 Detailed Description

Class for representing a XMLDictReader.

Note

bla Will read the DICOMV3.xml file

25.340.2 Constructor & Destructor Documentation

25.340.2.1 gdcm::XMLDictReader::XMLDictReader ( )

25.340.2.2 gdcm::XMLDictReader::~XMLDictReader ( ) [inline]

25.340.3 Member Function Documentation

25.340.3.1 void gdcm::XMLDictReader::CharacterDataHandler (const char * *data*, int *length*) [virtual]

Reimplemented from gdcm::TableReader.

25.340.3.2 void gdcm::XMLDictReader::EndElement (const char * *name*) [virtual]

Reimplemented from gdcm::TableReader.

25.340.3.3 const Dict& gdcm::XMLDictReader::GetDict ( ) [inline]

25.340.3.4 void gdcm::XMLDictReader::HandleDescription (const char ** *atts*) [protected]

25.340.3.5 void gdcm::XMLDictReader::HandleEntry (const char ** *atts*) [protected]

25.340.3.6 void gdcm::XMLDictReader::StartElement (const char * *name*, const char ** *atts*) [virtual]

Reimplemented from gdcm::TableReader.

The documentation for this class was generated from the following file:

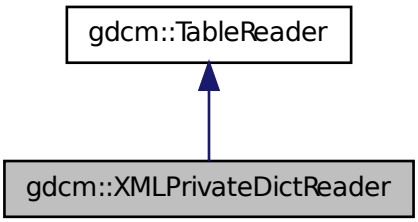
- gdcmXMLDictReader.h

25.341 gdcm::XMLPrivateDictReader Class Reference

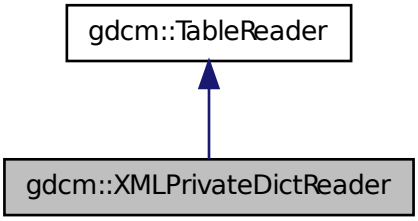
Class for representing a XMLPrivateDictReader.

```
#include <gdcmXMLPrivateDictReader.h>
```

Inheritance diagram for gdcm::XMLPrivateDictReader:



Collaboration diagram for gdcm::XMLPrivateDictReader:



Public Member Functions

- XMLPrivateDictReader ()
- ~XMLPrivateDictReader ()
- void CharacterDataHandler (const char *data, int length)
- void EndElement (const char *name)
- const PrivateDict & GetPrivateDict ()
- void StartElement (const char *name, const char **atts)

Protected Member Functions

- void HandleDescription (const char **atts)
- void HandleEntry (const char **atts)

25.341.1 Detailed Description

Class for representing a XMLPrivateDictReader.

Note

bla Will read the Private.xml file

25.341.2 Constructor & Destructor Documentation

25.341.2.1 gdcm::XMLPrivateDictReader::XMLPrivateDictReader ( )

25.341.2.2 gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ( ) [inline]

25.341.3 Member Function Documentation

25.341.3.1 void gdcm::XMLPrivateDictReader::CharacterDataHandler (const char * *data*, int *length*) [virtual]

Reimplemented from gdcm::TableReader.

25.341.3.2 void gdcm::XMLPrivateDictReader::EndElement (const char * *name*) [virtual]

Reimplemented from gdcm::TableReader.

25.341.3.3 const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ( ) [inline]

25.341.3.4 void gdcm::XMLPrivateDictReader::HandleDescription (const char ** *atts*) [protected]

25.341.3.5 void gdcm::XMLPrivateDictReader::HandleEntry (const char ** *atts*) [protected]

25.341.3.6 void gdcm::XMLPrivateDictReader::StartElement (const char * *name*, const char ** *atts*) [virtual]

Reimplemented from gdcm::TableReader.

The documentation for this class was generated from the following file:

- gdcmXMLPrivateDictReader.h

Chapter 26

File Documentation

26.1 gdcm2pnm.man File Reference

26.2 gdcm2vtk.man File Reference

26.3 gdcmAAbortPDU.h File Reference

Classes

- class gdcm::network::AAbortPDU

  AAbortPDU Table 9-26 A-ABORT PDU FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.4 gdcmAAssociateACPDU.h File Reference

Classes

- class gdcm::network::AAssociateACPDU

  AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.5 gdcmAAssociateRJPDU.h File Reference

Classes

- class gdcm::network::AAssociateRJPDU

  AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.6 gdcmAAssociateRQPDU.h File Reference

Classes

- class gdcm::network::AAssociateRQPDU

  AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.7 gdcmAbstractSyntax.h File Reference

Classes

- class gdcm::network::AbstractSyntax

  AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.8 gdcmanon.man File Reference

26.9 gdcmAnonymizeEvent.h File Reference

Classes

- class gdcm::AnonymizeEvent

  AnonymizeEvent Special type of event triggered during the Anonymization process.

Namespaces

- namespace gdcm

26.10 gdcmAnonymizer.h File Reference

Classes

- class gdcm::Anonymizer

  Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:

Namespaces

- namespace gdcm

26.11 gdcmApplicationContext.h File Reference

Classes

- class gdcm::network::ApplicationContext

  ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.12 gdcmApplicationEntity.h File Reference

Classes

- class gdcm::ApplicationEntity

  ApplicationEntity.

Namespaces

- namespace gdcm

26.13 gdcmAReleaseRPPDU.h File Reference

Classes

- class gdcm::network::AReleaseRPPDU

  AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.14 gdcmAReleaseRQPDU.h File Reference

Classes

- class gdcm::network::AReleaseRQPDU

  AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.15 gdcmARTIMTimer.h File Reference

Classes

- class gdcm::network::ARTIMTimer

  ARTIMTimer This file contains the code for the ARTIM timer.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.16 gdcmASN1.h File Reference

Classes

- class gdcm::ASN1

  Class for ASN1.

Namespaces

- namespace gdcm

26.17 gdcmAsynchronousOperationsWindowSub.h File Reference

Classes

- class gdcm::network::AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.18 gdcmAttribute.h File Reference

Classes

- class gdcm::Attribute< Group, Element, TVR, TVM >

  Attribute class This class use template metaprograming tricks to let the user know when the template instanciation does not match the public dictionary.
- class gdcm::Attribute< Group, Element, TVR, VM::VM1 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM2_n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM3_n >
- class gdcm::VRVLSize< 0 >
- class gdcm::VRVLSize< 1 >

Namespaces

- namespace gdcm

26.19 gdcmAudioCodec.h File Reference

Classes

- class gdcm::AudioCodec

  AudioCodec.

Namespaces

- namespace gdcm

26.20 gdcmBase64.h File Reference

Classes

- class gdcm::Base64

  Class for Base64.

Namespaces

- namespace gdcm

26.21 gdcmBaseCompositeMessage.h File Reference

Classes

- class gdcm::network::BaseCompositeMessage

  BaseCompositeMessage The Composite events described in section 3.7-2009 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2009 of the standard, and then fill in appropriate values in their datasets.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.22 gdcmBasePDU.h File Reference

Classes

- class gdcm::network::BasePDU

  BasePDU base class for PDUs.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.23 gdcmBaseRootQuery.h File Reference

Classes

- class gdcm::BaseRootQuery

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::EQueryLevel {
  gdcm::ePatient,
  gdcm::eStudy,
  gdcm::eSeries,
  gdcm::eImageOrFrame }

  BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root name and date: 18 oct 2010 mmr.
- enum gdcm::EQueryType {
  gdcm::eFind,
  gdcm::eMove }

26.24 gdcmBasicOffsetTable.h File Reference

Classes

- class gdcm::BasicOffsetTable

  Class to represent a BasicOffsetTable.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const BasicOffsetTable &val)

26.25 gdcmBitmap.h File Reference

Classes

- class gdcm::Bitmap

  Bitmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

Namespaces

- namespace gdcm

26.26 gdcmBitmapToBitmapFilter.h File Reference

Classes

- class gdcm::BitmapToBitmapFilter

  BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.

Namespaces

- namespace gdcm

26.27 gdcmByteBuffer.h File Reference

Classes

- class gdcm::ByteBuffer

  ByteBuffer.

Namespaces

- namespace gdcm

26.28 gdcmByteSwap.h File Reference

Classes

- class gdcm::ByteSwap< T >

  ByteSwap.

Namespaces

- namespace gdcm

26.29 gdcmByteSwapFilter.h File Reference

Classes

- class gdcm::ByteSwapFilter

  ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??

Namespaces

- namespace gdcm

26.30 gdcmByteValue.h File Reference

Classes

- class gdcm::ByteValue

  Class to represent binary value (array of bytes)

Namespaces

- namespace gdcm

26.31 gdcmCEchoMessages.h File Reference

Classes

- class gdcm::network::CEchoRQ

  CEchoRQ this file defines the messages for the cecho action.
- class gdcm::network::CEchoRSP

Namespaces

- namespace gdcm
- namespace gdcm::network

26.32 gdcmCFindMessages.h File Reference

Classes

- class gdcm::network::CFindCancelRQ
- class gdcm::network::CFindRQ
- class gdcm::network::CFindRSP

Namespaces

- namespace gdcm
- namespace gdcm::network

26.33 gdcmCMoveMessages.h File Reference

Classes

- class gdcm::network::CMoveCancelRq
- class gdcm::network::CMoveRQ

  CMoveRQ this file defines the messages for the cmove action.
- class gdcm::network::CMoveRSP

  CMoveRSP this file defines the messages for the cmove action.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.34 gdcmCodec.h File Reference

Classes

- class gdcm::Codec

  Codec class.

Namespaces

- namespace gdcm

26.35 gdcmCoder.h File Reference

Classes

- class gdcm::Coder

  Coder.

Namespaces

- namespace gdcm

26.36 gdcmCodeString.h File Reference

Classes

- class gdcm::CodeString

  CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

Namespaces

- namespace gdcm

Functions

- bool gdcm::operator!= (const CodeString &ref, const CodeString &cs)
- std::ostream & gdcm::operator<< (std::ostream &os, const CodeString &str)
- bool gdcm::operator== (const CodeString &ref, const CodeString &cs)

26.37 gdcmCommand.h File Reference

Classes

- class gdcm::Command

  Command superclass for callback/observer methods.
- class gdcm::MemberCommand< T >

  Command subclass that calls a pointer to a member function.
- class gdcm::SimpleMemberCommand< T >

  Command subclass that calls a pointer to a member function.

Namespaces

- namespace gdcm

26.38 gdcmCommandDataSet.h File Reference

Classes

- class gdcm::CommandDataSet

  Class to represent a Command DataSet.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CommandDataSet &val)

26.39 gdcmCompositeMessageFactory.h File Reference

Classes

- class gdcm::network::CompositeMessageFactory

  CompositeMessageFactory This class constructs PDataPDUs, but that have been specifically constructed for the composite DICOM services (C-Echo, C-Find, C-Get, C-Move, and C-Store). It will also handle parsing the incoming data to determine which of the CompositePDUs the incoming data is, and so therefore allowing the scu to determine what to do with incoming data (if acting as a storescp server, for instance).

Namespaces

- namespace gdcm
- namespace gdcm::network

26.40 gdcmCompositeNetworkFunctions.h File Reference

Classes

- class gdcm::CompositeNetworkFunctions

  Composite Network Functions These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

Namespaces

- namespace gdcm

26.41 gdcmConstCharWrapper.h File Reference

Classes

- class gdcm::ConstCharWrapper

  Do not use me.

Namespaces

- namespace gdcm

26.42 gdcmconv.man File Reference

26.43 gdcmCP246ExplicitDataElement.h File Reference

Classes

- class gdcm::CP246ExplicitDataElement

  Class to read/write a DataElement as CP246Explicit Data Element.

Namespaces

- namespace gdcm

26.44 gdcmCryptographicMessageSyntax.h File Reference

Classes

- class gdcm::CryptographicMessageSyntax

Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

Namespaces

- namespace gdcm

26.45 gdcmCSAElement.h File Reference

Classes

- class gdcm::CSAElement

  Class to represent a CSA Element.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)

26.46 gdcmCSAHeader.h File Reference

Classes

- class gdcm::CSAHeader

  Class for CSAHeader.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeader &d)

26.47 gdcmCSAHeaderDict.h File Reference

Classes

- class gdcm::CSAHeaderDict

  Class to represent a map of CSAHeaderDictEntry.
- class gdcm::CSAHeaderDictException

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)

26.48 gdcmCSAHeaderDictEntry.h File Reference

Classes

- class gdcm::CSAHeaderDictEntry

  Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)

26.49 gdcmCStoreMessages.h File Reference

Classes

- class gdcm::network::CStoreRQ

  CStoreRQ this file defines the messages for the cecho action.
- class gdcm::network::CStoreRSP

Namespaces

- namespace gdcm
- namespace gdcm::network

26.50 gdcmCurve.h File Reference

Classes

- class gdcm::Curve

  Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

Namespaces

- namespace gdcm

26.51 gdcmDataElement.h File Reference

Classes

- class gdcm::DataElement

  Class to represent a Data Element either Implicit or Explicit.

Namespaces

- namespace gdcm

Functions

- bool gdcm::operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & gdcm::operator<< (std::ostream &os, const DataElement &val)

26.52 gdcmDataEvent.h File Reference

Classes

- class gdcm::DataEvent

  DataEvent.

Namespaces

- namespace gdcm

26.53 gdcmDataSet.h File Reference

Classes

- class gdcm::DataElementException
- class gdcm::DataSet

  Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)

26.54 gdcmDataSetEvent.h File Reference

Classes

- class gdcm::DataSetEvent

  DataSetEvent Special type of event triggered during the DataSet store/move process.

Namespaces

- namespace gdcm

26.55 gdcmDataSetHelper.h File Reference

Classes

- class gdcm::DataSetHelper

  DataSetHelper (internal class, not intended for user level)

Namespaces

- namespace gdcm

26.56 gdcmDecoder.h File Reference

Classes

- class gdcm::Decoder

  Decoder.

Namespaces

- namespace gdcm

26.57 gdcmDefinedTerms.h File Reference

Classes

- class gdcm::DefinedTerms

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.

Namespaces

- namespace gdcm

26.58 gdcmDeflateStream.h File Reference

26.59 gdcmDefs.h File Reference

Classes

- class gdcm::Defs

  FIXME I do not like the name 'Defs'.

Namespaces

- namespace gdcm

26.60 gdcmDeltaEncodingCodec.h File Reference

Classes

- class gdcm::DeltaEncodingCodec

  DeltaEncodingCodec compression used by some private vendor.

Namespaces

- namespace gdcm

26.61 gdcmDICOMDIR.h File Reference

Classes

- class gdcm::DICOMDIR

  DICOMDIR class.

Namespaces

- namespace gdcm

26.62 gdcmDICOMDIRGenerator.h File Reference

Classes

- class gdcm::DICOMDIRGenerator

  DICOMDIRGenerator class This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.

Namespaces

- namespace gdcm

26.63 gdcmDict.h File Reference

Classes

- class gdcm::Dict

  Class to represent a map of DictEntry.
- class gdcm::PrivateDict

  Private Dict.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)
- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)

26.64 gdcmDictConverter.h File Reference

Classes

- class gdcm::DictConverter

  Class to convert a .dic file into something else:

Namespaces

- namespace gdcm

26.65 gdcmDictEntry.h File Reference

Classes

- class gdcm::DictEntry

  Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DictEntry &val)

26.66 gdcmDictPrinter.h File Reference

Classes

- class gdcm::DictPrinter

  DictPrinter class.

Namespaces

- namespace gdcm

26.67 gdcmDicts.h File Reference

Classes

- class gdcm::Dicts

  Class to manipulate the sum of knowledge (all the dict user load)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)

26.68 gdcmdiff.man File Reference

26.69 gdcmDIMSE.h File Reference

Classes

- class gdcm::network::CEchoRQ

  CEchoRQ this file defines the messages for the cecho action.
- class gdcm::network::CEchoRSP
- class gdcm::network::CFind
- class gdcm::network::DIMSE

  DIMSE PS 3.7 - 2009 Annex E Command Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS Table E.1-1 COMMAND FIELDS (PART 1)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.70 gdcmDirectionCosines.h File Reference

Classes

- class gdcm::DirectionCosines

  class to handle DirectionCosines

Namespaces

- namespace gdcm

26.71 gdcmDirectory.h File Reference

Classes

- class gdcm::Directory

  Class for manipulation directories.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Directory &d)

26.72 gdcmDirectoryHelper.h File Reference

Classes

- class gdcm::DirectoryHelper

Namespaces

- namespace gdcm

26.73 gdcmDummyValueGenerator.h File Reference

Classes

- class gdcm::DummyValueGenerator

  Class for generating dummy value.

Namespaces

- namespace gdcm

26.74 gdcmdump.man File Reference

26.75 gdcmDumper.h File Reference

Classes

- class gdcm::Dumper

  Codec class.

Namespaces

- namespace gdcm

26.76 gdcmElement.h File Reference

Classes

- class gdcm::Element< TVR, TVM >

  Element class.
- class gdcm::Element< TVR, VM::VM1_2 >
- class gdcm::Element< TVR, VM::VM1_n >
- class gdcm::Element< TVR, VM::VM2_2n >
- class gdcm::Element< TVR, VM::VM2_n >

- class gdcm::Element< TVR, VM::VM3_3n >
- class gdcm::Element< TVR, VM::VM3_n >
- class gdcm::Element< VR::AS, VM::VM5 >
- class gdcm::Element< VR::OB, VM::VM1 >
- class gdcm::Element< VR::OW, VM::VM1 >
- class gdcm::EncodingImplementation< VR::VRASCII >
- class gdcm::EncodingImplementation< VR::VRBINARY >
- struct gdcm::ignore_char

Namespaces

- namespace gdcm

Functions

- ignore_char const gdcm::backslash ('\\')
- std::istream & gdcm::operator>> (std::istream &in, ignore_char const &ic)
- template<typename Float >
  std::string gdcm::to_string (Float data)

26.77 gdcmEncapsulatedDocument.h File Reference

Classes

- class gdcm::EncapsulatedDocument

  EncapsulatedDocument.

Namespaces

- namespace gdcm

26.78 gdcmEnumeratedValues.h File Reference

Classes

- class gdcm::EnumeratedValues

  Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

Namespaces

- namespace gdcm

26.79 gdcmEvent.h File Reference

Classes

- class gdcm::AbortEvent
- class gdcm::AnyEvent
- class gdcm::EndEvent
- class gdcm::Event

  superclass for callback/observer methods
- class gdcm::ExitEvent
- class gdcm::InitializeEvent
- class gdcm::IterationEvent
- class gdcm::ModifiedEvent
- class gdcm::NoEvent
- class gdcm::StartEvent
- class gdcm::UserEvent

Namespaces

- namespace gdcm

Macros

- #define gdcmEventMacro(classname, super)

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, Event &e)

  Generic inserter operator for Event and its subclasses.

26.79.1 Macro Definition Documentation

26.79.1.1 #define gdcmEventMacro(*classname, super*)

Value:

```
 \
class  classname : public super { \
  public: \
    typedef classname Self; \
    typedef super Superclass; \
    classname() {} \
    virtual ~classname() {} \
    virtual const char * GetEventName() const { return #classname; } \
    virtual bool CheckEvent(const ::gdcm::Event* e) const \
      { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcm::Event* MakeObject() const \
      { return new Self; } \
    classname(const Self&s) : super(s){}; \
  private: \
    void operator=(const Self&); \
}
```

26.80 gdcmException.h File Reference

Classes

- class gdcm::Exception

  Exception.

Namespaces

- namespace gdcm

26.81 gdcmExplicitDataElement.h File Reference

Classes

- class gdcm::ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.

Namespaces

- namespace gdcm

26.82 gdcmExplicitImplicitDataElement.h File Reference

Classes

- class gdcm::ExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element.

Namespaces

- namespace gdcm

26.83 gdcmFiducials.h File Reference

Classes

- class gdcm::Fiducials

  Fiducials.

Namespaces

- namespace gdcm

26.84 gdcmFile.h File Reference

Classes

- class gdcm::File

  a DICOM File See PS 3.10 File: A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may by written, read and/or deleted.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const File &val)

26.85 gdcmFileDerivation.h File Reference

Classes

- class gdcm::FileDerivation

  FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.

Namespaces

- namespace gdcm

26.86 gdcmFileExplicitFilter.h File Reference

Classes

- class gdcm::FileExplicitFilter

  FileExplicitFilter class After changing a file from Implicit to Explicit representation (see ImageChangeTransferSyntax) one operation is to make sure the VR of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit reprensentation, the VR is not stored directly in the file.

Namespaces

- namespace gdcm

26.87 gdcmFileMetaInformation.h File Reference

Classes

- class gdcm::FileMetaInformation

  Class to represent a File Meta Information.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileMetaInformation &val)

26.88 gdcmFilename.h File Reference

Classes

- class gdcm::Filename

  Class to manipulate file name's.

Namespaces

- namespace gdcm

26.89 gdcmFilenameGenerator.h File Reference

Classes

- class gdcm::FilenameGenerator

  FilenameGenerator.

Namespaces

- namespace gdcm

26.90 gdcmFileSet.h File Reference

Classes

- class gdcm::FileSet

  File-set: A File-set is a collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which File IDs are unique.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)

26.91 gdcmFindPatientRootQuery.h File Reference

Classes

- class gdcm::FindPatientRootQuery

  PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

Namespaces

- namespace gdcm

26.92 gdcmFindStudyRootQuery.h File Reference

Classes

- class gdcm::FindStudyRootQuery

  FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

Namespaces

- namespace gdcm

26.93 gdcmFragment.h File Reference

Classes

- class gdcm::Fragment

  Class to represent a Fragment.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Fragment &val)

26.94 gdcmgendir.man File Reference

26.95 gdcmGlobal.h File Reference

Classes

- class gdcm::Global

  Global.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)

Variables

- static Global gdcm::GlobalInstance

26.96 gdcmGroupDict.h File Reference

Classes

- class gdcm::GroupDict

  Class to represent the mapping from group number to its abbreviation and name.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)

26.97 gdcmIconImage.h File Reference

Namespaces

- namespace gdcm

Typedefs

- typedef Bitmap gdcm::IconImage

26.98 gdcmIconImageFilter.h File Reference

Classes

- class gdcm::IconImageFilter

  IconImageFilter This filter will extract icons from a gdcm::File This filter will loop over all known sequence (public and private) that may contains an IconImage and retrieve them. The filter will fails with a value of false if no icon can be found Since it handle both public and private icon type, one should not assume the icon is in uncompress form, some private vendor store private icon in JPEG8/JPEG12.

Namespaces

- namespace gdcm

26.99 gdcmIconImageGenerator.h File Reference

Classes

- class gdcm::IconImageGenerator

  IconImageGenerator This filter will generate a valid Icon from the Pixel Data element (an instance of gdcm::Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

Namespaces

- namespace gdcm

26.100 gdcmImage.h File Reference

Classes

- class gdcm::Image

  Image.

Namespaces

- namespace gdcm

26.101 gdcmImageApplyLookupTable.h File Reference

Classes

- class gdcm::ImageApplyLookupTable

  ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

Namespaces

- namespace gdcm

26.102 gdcmImageChangePhotometricInterpretation.h File Reference

Classes

- class gdcm::ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.

Namespaces

- namespace gdcm

26.103 gdcmImageChangePlanarConfiguration.h File Reference

Classes

- class gdcm::ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.

Namespaces

- namespace gdcm

26.104 gdcmImageChangeTransferSyntax.h File Reference

Classes

- class gdcm::ImageChangeTransferSyntax

  ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.

Namespaces

- namespace gdcm

26.105 gdcmImageCodec.h File Reference

Classes

- class gdcm::ImageCodec

  ImageCodec.

Namespaces

- namespace gdcm

26.106 gdcmImageConverter.h File Reference

Classes

- class gdcm::ImageConverter

  Image Converter.

Namespaces

- namespace gdcm

26.107 gdcmImageFragmentSplitter.h File Reference

Classes

- class gdcm::ImageFragmentSplitter

  ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

Namespaces

- namespace gdcm

26.108 gdcmImageHelper.h File Reference

Classes

- class gdcm::ImageHelper

  ImageHelper (internal class, not intended for user level)

Namespaces

- namespace gdcm

26.109 gdcmImageReader.h File Reference

Classes

- class gdcm::ImageReader

  ImageReader.

Namespaces

- namespace gdcm

26.110 gdcmImageToImageFilter.h File Reference

Classes

- class gdcm::ImageToImageFilter

  ImageToImageFilter class Super class for all filter taking an image and producing an output image.

Namespaces

- namespace gdcm

26.111 gdcmImageWriter.h File Reference

Classes

- class gdcm::ImageWriter

  ImageWriter.

Namespaces

- namespace gdcm

26.112 gdcmimg.man File Reference

26.113 gdcmImplementationClassUIDSub.h File Reference

Classes

- class gdcm::network::ImplementationClassUIDSub

  ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE--RQ)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.114 gdcmImplementationUIDSub.h File Reference

Classes

- class gdcm::network::ImplementationUIDSub

  ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.115 gdcmImplementationVersionNameSub.h File Reference

Classes

- class gdcm::network::ImplementationVersionNameSub

  ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.116 gdcmImplicitDataElement.h File Reference

Classes

- class gdcm::ImplicitDataElement

  Class to represent an Implicit VR Data Element.

Namespaces

- namespace gdcm

26.117 gdcminfo.man File Reference

26.118 gdcmIOD.h File Reference

Classes

- class gdcm::IOD

  Class for representing a IOD.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)

26.119 gdcmIODEntry.h File Reference

Classes

- class gdcm::IODEntry

  Class for representing a IODEntry.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)

26.120 gdcmIODs.h File Reference

Classes

- class gdcm::IODs

  Class for representing a IODs.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)

26.121 gdcmIPPSorter.h File Reference

Classes

- class gdcm::IPPSorter

  IPPSorter Implement a simple Image Position (Patient) sorter, along the Image Orientation (Patient) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

Namespaces

- namespace gdcm

26.122 gdcmItem.h File Reference

Classes

- class gdcm::Item

  Class to represent an Item A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standart Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways Explicit/ Implicit.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Item &val)

26.123 gdcmJPEG12Codec.h File Reference

Classes

- class gdcm::JPEG12Codec

  Class to do JPEG 12bits (lossy & lossless)

Namespaces

- namespace gdcm

26.124 gdcmJPEG16Codec.h File Reference

Classes

- class gdcm::JPEG16Codec

  Class to do JPEG 16bits (lossless)

Namespaces

- namespace gdcm

26.125 gdcmJPEG2000Codec.h File Reference

Classes

- class gdcm::JPEG2000Codec

  Class to do JPEG 2000.

Namespaces

- namespace gdcm

26.126 gdcmJPEG8Codec.h File Reference

Classes

- class gdcm::JPEG8Codec

  Class to do JPEG 8bits (lossy & lossless)

Namespaces

- namespace gdcm

26.127 gdcmJPEGCodec.h File Reference

Classes

- class gdcm::JPEGCodec

  JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: gdcm::JPEG8Codec, gdcm::JPEG12Codec & gdcm::JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case.

Namespaces

- namespace gdcm

26.128 gdcmJPEGLSCodec.h File Reference

Classes

- class gdcm::JPEGLSCodec
  JPEG-LS.

Namespaces

- namespace gdcm

26.129 gdcmKAKADUCodec.h File Reference

Classes

- class gdcm::KAKADUCodec
  KAKADUCodec.

Namespaces

- namespace gdcm

26.130 gdcmLegacyMacro.h File Reference

Macros

- #define GDCM_LEGACY(method) method;
- #define GDCM_LEGACY_BODY(method, version) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")
- #define GDCM_LEGACY_REPLACED_BODY(method, version, replace) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version. Use " #replace " instead.")

26.130.1 Macro Definition Documentation

26.130.1.1 #define GDCM_LEGACY(*method*) method;

26.130.1.2 #define GDCM_LEGACY_BODY(*method, version*) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")

26.130.1.3 #define GDCM_LEGACY_REPLACED_BODY(*method, version, replace*) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version. Use " #replace " instead.")

26.131 gdcmLO.h File Reference

Classes

- class gdcm::LO

LO.

Namespaces

- namespace gdcm

26.132 gdcmLookupTable.h File Reference

Classes

- class gdcm::LookupTable
  LookupTable class.

Namespaces

- namespace gdcm

26.133 gdcmMacro.h File Reference

Classes

- class gdcm::Macro
  Class for representing a Macro.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)

26.134 gdcmMacroEntry.h File Reference

Macros

- #define GDCMMACROENTRY_H

26.134.1 Macro Definition Documentation

26.134.1.1 #define GDCMMACROENTRY_H

26.135 gdcmMacros.h File Reference

Classes

- class gdcm::Macros

  Class for representing a Modules.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)

26.136 gdcmMaximumLengthSub.h File Reference

Classes

- class gdcm::network::MaximumLengthSub

  MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.137 gdcmMD5.h File Reference

Classes

- class gdcm::MD5

  Class for MD5.

Namespaces

- namespace gdcm

26.138 gdcmMediaStorage.h File Reference

Classes

- class gdcm::MediaStorage

  MediaStorage.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)

26.139 gdcmMeshPrimitive.h File Reference

Classes

- class gdcm::MeshPrimitive

  This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

Namespaces

- namespace gdcm

26.140 gdcmModule.h File Reference

Classes

- class gdcm::Module

  Class for representing a Module.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)

26.141 gdcmModuleEntry.h File Reference

Classes

- class gdcm::ModuleEntry

  Class for representing a ModuleEntry.

Namespaces

- namespace gdcm

Typedefs

- typedef ModuleEntry gdcm::MacroEntry

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const ModuleEntry &_val)

26.142 gdcmModules.h File Reference

Classes

- class gdcm::Modules

  Class for representing a Modules.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)

26.143 gdcmMovePatientRootQuery.h File Reference

Classes

- class gdcm::MovePatientRootQuery

  MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.

Namespaces

- namespace gdcm

26.144 gdcmMoveStudyRootQuery.h File Reference

Classes

- class gdcm::MoveStudyRootQuery

  MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

Namespaces

- namespace gdcm

26.145 gdcmNestedModuleEntries.h File Reference

Classes

- class gdcm::NestedModuleEntries

  Class for representing a NestedModuleEntries.

Namespaces

- namespace gdcm

Typedefs

- typedef NestedModuleEntries gdcm::NestedMacroEntries

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const NestedModuleEntries &_val)

26.146 gdcmNetworkEvents.h File Reference

Namespaces

- namespace gdcm
- namespace gdcm::network

Enumerations

- enum gdcm::network::EEventID {
  gdcm::network::eAASSOCIATERequestLocalUser = 0,
  gdcm::network::eTransportConnConfirmLocal,
  gdcm::network::eASSOCIATE_ACPDUreceived,
  gdcm::network::eASSOCIATE_RJPDUreceived,
  gdcm::network::eTransportConnIndicLocal,
  gdcm::network::eAASSOCIATE_RQPDUreceived,
  gdcm::network::eAASSOCIATEresponseAccept,
  gdcm::network::eAASSOCIATEresponseReject,
  gdcm::network::ePDATArequest,
  gdcm::network::ePDATATFPDU,
  gdcm::network::eARELEASERequest,
  gdcm::network::eARELEASE_RQPDUReceivedOpen,
  gdcm::network::eARELEASE_RPPDUReceived,
  gdcm::network::eARELEASEResponse,
  gdcm::network::eAABORTRequest,
  gdcm::network::eAABORTPDUReceivedOpen,
  gdcm::network::eTransportConnectionClosed,
  gdcm::network::eARTIMTimerExpired,
  gdcm::network::eUnrecognizedPDUReceived,
  gdcm::network::eEventDoesNotExist }

Variables

- const int gdcm::network::cMaxEventID = eEventDoesNotExist

26.147 gdcmNetworkStateID.h File Reference

Namespaces

- namespace gdcm
- namespace gdcm::network

Enumerations

- enum gdcm::network::EStateID {
  gdcm::network::eStaDoesNotExist = 0,
  gdcm::network::eSta1Idle = 1,
  gdcm::network::eSta2Open = 2,
  gdcm::network::eSta3WaitLocalAssoc = 4,
  gdcm::network::eSta4LocalAssocDone = 8,
  gdcm::network::eSta5WaitRemoteAssoc = 16,
  gdcm::network::eSta6TransferReady = 32,
  gdcm::network::eSta7WaitRelease = 64,
  gdcm::network::eSta8WaitLocalRelease = 128,
  gdcm::network::eSta9ReleaseCollisionRqLocal = 256,
  gdcm::network::eSta10ReleaseCollisionAc = 512,
  gdcm::network::eSta11ReleaseCollisionRq = 1024,
  gdcm::network::eSta12ReleaseCollisionAcLocal = 2048,
  gdcm::network::eSta13AwaitingClose = 4096 }

Functions

- int gdcm::network::GetStateIndex (EStateID inState)

Variables

- const int gdcm::network::cMaxStateID = 13

26.148 gdcmObject.h File Reference

Classes

- class gdcm::Object

  Object.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)

26.149 gdcmOrientation.h File Reference

Classes

- class gdcm::Orientation

  class to handle Orientation

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)

26.150 gdcmOverlay.h File Reference

Classes

- class gdcm::Overlay

  Overlay class.

Namespaces

- namespace gdcm

26.151 gdcmParseException.h File Reference

Classes

- class gdcm::ParseException

  ParseException Standard exception handling object.

Namespaces

- namespace gdcm

26.152 gdcmParser.h File Reference

Classes

- class gdcm::Parser

  Parser ala XML_Parser from expat (SAX)

Namespaces

- namespace gdcm

26.153 gdcmPatient.h File Reference

Classes

- class gdcm::Patient

  See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

Namespaces

- namespace gdcm

26.154 gdcmPDataTFPDU.h File Reference

Classes

- class gdcm::network::PDataTFPDU

  PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.155 gdcmPDBElement.h File Reference

Classes

- class gdcm::PDBElement

  Class to represent a PDB Element.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PDBElement &val)

26.156 gdcmPDBHeader.h File Reference

Classes

- class gdcm::PDBHeader

  Class for PDBHeader.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PDBHeader &d)

26.157 gdcmpdf.man File Reference

26.158 gdcmPDFCodec.h File Reference

Classes

- class gdcm::PDFCodec

  PDFCodec class.

Namespaces

- namespace gdcm

26.159 gdcmPDUFactory.h File Reference

Classes

- class gdcm::network::PDUFactory

  PDUFactory basically, given an initial byte, construct the appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.160 gdcmPersonName.h File Reference

Classes

- class gdcm::PersonName

  PersonName class.

Namespaces

- namespace gdcm

26.161 gdcmPhotometricInterpretation.h File Reference

Classes

- class gdcm::PhotometricInterpretation

  Class to represent an PhotometricInterpretation.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)

26.162 gdcmPixelFormat.h File Reference

Classes

- class gdcm::PixelFormat

  PixelFormat.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)

26.163 gdcmPixmap.h File Reference

Classes

- class gdcm::Pixmap

  Pixmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

Namespaces

- namespace gdcm

26.164 gdcmPixmapReader.h File Reference

Classes

- class gdcm::PixmapReader

  PixmapReader.

Namespaces

- namespace gdcm

26.165 gdcmPixmapToPixmapFilter.h File Reference

Classes

- class gdcm::PixmapToPixmapFilter

  PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.

Namespaces

- namespace gdcm

26.166 gdcmPixmapWriter.h File Reference

Classes

- class gdcm::PixmapWriter

  PixmapWriter This class will takes two inputs:

Namespaces

- namespace gdcm

26.167 gdcmPNMCodec.h File Reference

Classes

- class gdcm::PNMCodec

  Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/`.

Namespaces

- namespace gdcm

26.168 gdcmPreamble.h File Reference

Classes

- class gdcm::Preamble

  DICOM Preamble (Part 10)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)

26.169 gdcmPresentationContext.h File Reference

Classes

- class gdcm::PresentationContext

  PresentationContext.

Namespaces

- namespace gdcm

26.170 gdcmPresentationContextAC.h File Reference

Classes

- class gdcm::network::PresentationContextAC

  PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.171 gdcmPresentationContextGenerator.h File Reference

Classes

- class gdcm::PresentationContextGenerator

  PresentationContextGenerator This class is responsible for generating the proper PresentationContext that will be used in subsequent operation during a DICOM Query/Retrieve association. The step of the association is very sensible as special care need to be taken to explicitly define what instance are going to be send and how they are encoded.

Namespaces

- namespace gdcm

26.172 gdcmPresentationContextRQ.h File Reference

Classes

- class gdcm::network::PresentationContextRQ

  PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.173 gdcmPresentationDataValue.h File Reference

Classes

- class gdcm::network::PresentationDataValue

  PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.174 gdcmPrinter.h File Reference

Classes

- class gdcm::Printer

  Printer class.

Namespaces

- namespace gdcm

26.175 gdcmPrivateTag.h File Reference

Classes

- class gdcm::PrivateTag

  Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)

26.176 gdcmProgressEvent.h File Reference

Classes

- class gdcm::ProgressEvent

  ProgressEvent Special type of event triggered during.

Namespaces

- namespace gdcm

26.177 gdcmPVRGCodec.h File Reference

Classes

- class gdcm::PVRGCodec

  PVRGCodec.

Namespaces

- namespace gdcm

26.178 gdcmPythonFilter.h File Reference

Classes

- class gdcm::PythonFilter

  PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

Namespaces

- namespace gdcm

26.179 gdcmQueryBase.h File Reference

Classes

- class gdcm::QueryBase

  QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ERootType {
  gdcm::ePatientRootType,
  gdcm::eStudyRootType }

26.180 gdcmQueryFactory.h File Reference

Classes

- class gdcm::QueryFactory

  QueryFactory.h.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ECharSet {
  gdcm::eLatin1 = 0,
  gdcm::eLatin2,
  gdcm::eLatin3,
  gdcm::eLatin4,
  gdcm::eCyrillic,
  gdcm::eArabic,
  gdcm::eGreek,
  gdcm::eHebrew,
  gdcm::eLatin5,
  gdcm::eJapanese,
  gdcm::eThai,
  gdcm::eJapaneseKanjiMultibyte,
  gdcm::eJapaneseSupplementaryKanjiMultibyte,
  gdcm::eKoreanHangulHanjaMultibyte,
  gdcm::eUTF8,
  gdcm::eGB18030 }

26.181 gdcmQueryImage.h File Reference

Classes

- class gdcm::QueryImage

  QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

Namespaces

- namespace gdcm

26.182 gdcmQueryPatient.h File Reference

Classes

- class gdcm::QueryPatient

  QueryPatient contains: class to construct a patient-based query for c-find and c-move.

Namespaces

- namespace gdcm

26.183 gdcmQuerySeries.h File Reference

Classes

- class gdcm::QuerySeries

QuerySeries contains: class to construct a series-based query for c-find and c-move.

Namespaces

- namespace gdcm

26.184 gdcmQueryStudy.h File Reference

Classes

- class gdcm::QueryStudy

  QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

Namespaces

- namespace gdcm

26.185 gdcmraw.man File Reference

26.186 gdcmRAWCodec.h File Reference

Classes

- class gdcm::RAWCodec

  RAWCodec class.

Namespaces

- namespace gdcm

26.187 gdcmReader.h File Reference

Classes

- class gdcm::Reader

  Reader ala DOM (Document Object Model)

Namespaces

- namespace gdcm

26.188 gdcmRescaler.h File Reference

Classes

- class gdcm::Rescaler

  Rescale class This class is meant to apply the linear transform of Stored Pixel Value to Real World Value. This is mostly found in CT or PET dataset, where the value are stored using one type, but need to be converted to another scale using a linear transform. There are basically two cases: In CT: the linear transform is generally integer based. E.g. the Stored Pixel Type is unsigned short 12bits, but to get Hounsfield unit, one need to apply the linear transform:

  $$RWV = 1. * SV - 1024$$

  So the best scalar to store the Real World Value will be 16 bits signed type.

Namespaces

- namespace gdcm

26.189 gdcmRLECodec.h File Reference

Classes

- class gdcm::RLECodec

  Class to do RLE.

Namespaces

- namespace gdcm

26.190 gdcmScanner.h File Reference

Classes

- struct gdcm::Scanner::ltstr
- class gdcm::Scanner

  Scanner This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)

26.191 gdcmscanner.man File Reference

26.192 gdcmscu.man File Reference

26.193 gdcmSegment.h File Reference

Classes

- class gdcm::Segment

  This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

Namespaces

- namespace gdcm

26.194 gdcmSegmentedPaletteColorLookupTable.h File Reference

Classes

- class gdcm::SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.

Namespaces

- namespace gdcm

26.195 gdcmSegmentHelper.h File Reference

Classes

- struct gdcm::SegmentHelper::BasicCodedEntry

  This structure defines a basic coded entry with all of its attributes.

Namespaces

- namespace gdcm
- namespace gdcm::SegmentHelper

26.196 gdcmSegmentReader.h File Reference

Classes

- class gdcm::SegmentReader

  This class defines a segment reader. It reads attributes of group 0x0062.

Namespaces

- namespace gdcm

26.197 gdcmSegmentWriter.h File Reference

Classes

- class gdcm::SegmentWriter

  This class defines a segment writer. It writes attributes of group 0x0062.

Namespaces

- namespace gdcm

26.198 gdcmSequenceOfFragments.h File Reference

Classes

- class gdcm::SequenceOfFragments

  Class to represent a Sequence Of Fragments.

Namespaces

- namespace gdcm

26.199 gdcmSequenceOfItems.h File Reference

Classes

- class gdcm::SequenceOfItems

  Class to represent a Sequence Of Items (value representation : SQ)

Namespaces

- namespace gdcm

26.200 gdcmSerieHelper.h File Reference

Classes

- class gdcm::FileWithName

  FileWithName.
- struct gdcm::SerieHelper::Rule
- class gdcm::SerieHelper

Namespaces

- namespace gdcm

Typedefs

- typedef bool(∗ gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER)(File ∗, File ∗)
- typedef std::vector
  < SmartPointer< FileWithName > > gdcm::FileList

Enumerations

- enum gdcm::CompOperators {
  gdcm::GDCM_EQUAL = 0,
  gdcm::GDCM_DIFFERENT,
  gdcm::GDCM_GREATER,
  gdcm::GDCM_GREATEROREQUAL,
  gdcm::GDCM_LESS,
  gdcm::GDCM_LESSOREQUAL }
- enum gdcm::LodModeType {
  gdcm::LD_ALL = 0x00000000,
  gdcm::LD_NOSEQ = 0x00000001,
  gdcm::LD_NOSHADOW = 0x00000002,
  gdcm::LD_NOSHADOWSEQ = 0x00000004 }

26.201 gdcmSeries.h File Reference

Classes

- class gdcm::Series
  Series.

Namespaces

- namespace gdcm

26.202 gdcmServiceClassUser.h File Reference

Classes

- class gdcm::ServiceClassUser
  ServiceClassUser.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.203 gdcmSHA1.h File Reference

Classes

- class gdcm::SHA1

  Class for SHA1.

Namespaces

- namespace gdcm

26.204 gdcmSimpleSubjectWatcher.h File Reference

Classes

- class gdcm::SimpleSubjectWatcher

  SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events.

Namespaces

- namespace gdcm

26.205 gdcmSmartPointer.h File Reference

Classes

- class gdcm::SmartPointer< ObjectType >

  Class for Smart Pointer.

Namespaces

- namespace gdcm

26.206 gdcmSOPClassUIDToIOD.h File Reference

Classes

- class gdcm::SOPClassUIDToIOD

  Class convert a class SOP Class UID into IOD.

Namespaces

- namespace gdcm

26.207 gdcmSorter.h File Reference

Classes

- class gdcm::Sorter

  Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)

26.208 gdcmSpacing.h File Reference

Classes

- class gdcm::Spacing

  Class for Spacing.

Namespaces

- namespace gdcm

26.209 gdcmSpectroscopy.h File Reference

Classes

- class gdcm::Spectroscopy

  Spectroscopy class.

Namespaces

- namespace gdcm

26.210 gdcmSplitMosaicFilter.h File Reference

Classes

- class gdcm::SplitMosaicFilter

  SplitMosaicFilter class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA Image Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.

Namespaces

- namespace gdcm

26.211 gdcmStaticAssert.h File Reference

Classes

- struct gdcm::static_assert_test< x >
- struct gdcm::STATIC_ASSERTION_FAILURE< true >

Namespaces

- namespace gdcm

Macros

- #define GDCM_DO_JOIN(X, Y) GDCM_DO_JOIN2(X,Y)
- #define GDCM_DO_JOIN2(X, Y) X##Y
- #define GDCM_JOIN(X, Y) GDCM_DO_JOIN(X, Y)
- #define GDCM_STATIC_ASSERT(B)

  *The GDCM_JOIN + **LINE** is needed to create a uniq identifier.*

26.211.1 Macro Definition Documentation

26.211.1.1 #define GDCM_DO_JOIN(*X, Y*) GDCM_DO_JOIN2(X,Y)

26.211.1.2 #define GDCM_DO_JOIN2(*X, Y*) X##Y

26.211.1.3 #define GDCM_JOIN(*X, Y*) GDCM_DO_JOIN(X, Y)

26.211.1.4 #define GDCM_STATIC_ASSERT(*B*)

Value:

```
typedef ::gdcm::static_assert_test<\
    sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >)>\
      GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
```

The GDCM_JOIN + **LINE** is needed to create a uniq identifier.

26.212 gdcmStreamImageReader.h File Reference

Classes

- struct gdcm::OneShotReadBuf
- class gdcm::StreamImageReader

  StreamImageReader.

Namespaces

- namespace gdcm

26.213 gdcmStreamImageWriter.h File Reference

Classes

- class gdcm::StreamImageWriter

  StreamImageReader.

Namespaces

- namespace gdcm

26.214 gdcmString.h File Reference

Classes

- class gdcm::String< TDelimiter, TMaxLength, TPadChar >

  String.

Namespaces

- namespace gdcm

Functions

- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
  std::istream & gdcm::operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)

26.215 gdcmStringFilter.h File Reference

Classes

- class gdcm::StringFilter

  StringFilter StringFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

Namespaces

- namespace gdcm

26.216 gdcmStudy.h File Reference

Classes

- class gdcm::Study

  Study.

Namespaces

- namespace gdcm

26.217 gdcmSubject.h File Reference

Classes

- class gdcm::Subject

  Subject.

Namespaces

- namespace gdcm

26.218 gdcmSurface.h File Reference

Classes

- class gdcm::Surface

  This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

Namespaces

- namespace gdcm

26.219 gdcmSurfaceHelper.h File Reference

Classes

- class gdcm::SurfaceHelper

Namespaces

- namespace gdcm

26.220 gdcmSurfaceReader.h File Reference

Classes

- class gdcm::SurfaceReader

  This class defines a SURFACE IE reader. It reads surface mesh module attributes.

Namespaces

- namespace gdcm

26.221 gdcmSurfaceWriter.h File Reference

Classes

- class gdcm::SurfaceWriter

  This class defines a SURFACE IE writer. It writes surface mesh module attributes.

Namespaces

- namespace gdcm

26.222 gdcmSwapCode.h File Reference

Classes

- class gdcm::SwapCode

  SwapCode representation.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)

26.223 gdcmSwapper.h File Reference

Classes

- class gdcm::SwapperDoOp
- class gdcm::SwapperNoOp

Namespaces

- namespace gdcm

26.224 gdcmSystem.h File Reference

Classes

- class gdcm::System

  Class to do system operation.

Namespaces

- namespace gdcm

26.225 gdcmTable.h File Reference

Classes

- class gdcm::Table

  Table.

Namespaces

- namespace gdcm

26.226 gdcmTableEntry.h File Reference

Classes

- class gdcm::TableEntry

  TableEntry.

Namespaces

- namespace gdcm

26.227 gdcmTableReader.h File Reference

Classes

- class gdcm::TableReader

  Class for representing a TableReader.

Namespaces

- namespace gdcm

26.228 gdcmTag.h File Reference

Classes

- class gdcm::Tag

  Class to represent a DICOM Data Element (Attribute) Tag (Group, Element). Basically an uint32_t which can also be expressed as two uint16_t (group and element)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)
- std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)

26.229 gdcmTagPath.h File Reference

Classes

- class gdcm::TagPath

  class to handle a path of tag.

Namespaces

- namespace gdcm

26.230 gdcmTagToVR.h File Reference

Namespaces

- namespace gdcm

Functions

- VR::VRType gdcm::GetVRFromTag (Tag const &tag)

26.231 gdcmtar.man File Reference

26.232 gdcmTerminal.h File Reference

Namespaces

- namespace gdcm
- namespace gdcm::terminal

  Class for Terminal Allow one to print in color in a shell.

Enumerations

- enum gdcm::terminal::Attribute {
  gdcm::terminal::reset = 0,
  gdcm::terminal::bright = 1,
  gdcm::terminal::dim = 2,
  gdcm::terminal::underline = 3,
  gdcm::terminal::blink = 5,
  gdcm::terminal::reverse = 7,
  gdcm::terminal::hidden = 8 }
- enum gdcm::terminal::Color {
  gdcm::terminal::black = 0,
  gdcm::terminal::red,
  gdcm::terminal::green,
  gdcm::terminal::yellow,
  gdcm::terminal::blue,
  gdcm::terminal::magenta,
  gdcm::terminal::cyan,
  gdcm::terminal::white }
- enum gdcm::terminal::Mode {
  gdcm::terminal::CONSOLE = 0,
  gdcm::terminal::VT100 }

Functions

- GDCM_EXPORT std::string gdcm::terminal::setattribute (Attribute att)
- GDCM_EXPORT std::string gdcm::terminal::setbgcolor (Color c)
- GDCM_EXPORT std::string gdcm::terminal::setfgcolor (Color c)
- GDCM_EXPORT void gdcm::terminal::setmode (Mode m)

26.233 gdcmTestDriver.h File Reference

26.234 gdcmTesting.h File Reference

Classes

- class gdcm::Testing

  class for testing

Namespaces

- namespace gdcm

26.235 gdcmTrace.h File Reference

Classes

- class gdcm::Trace

  Trace.

Namespaces

- namespace gdcm

Macros

- #define GDCM_FUNCTION "<unknow>"
- #define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(arg)

  AssertAlways.
- #define gdcmAssertMacro(arg)

  Assert.
- #define gdcmDebugMacro(msg)

  Debug.
- #define gdcmErrorMacro(msg)

  Error this is pretty bad, more than just warning It could mean lost of data, something not handle...
- #define gdcmWarningMacro(msg)

  Warning.

26.235.1 Macro Definition Documentation

26.235.1.1 #define GDCM_FUNCTION "<unknow>"

26.235.1.2 #define gdcmAssertAlwaysMacro(*arg*) gdcmAssertMacro(arg)

AssertAlways.

Parameters

arg	argument to test An easy solution to pass also a message is to do: gdcmAssertMacro("my message" && 2 < 3)

Referenced by gdcm::VR::Write().

26.235.1.3 #define gdcmAssertMacro(*arg*)

Value:

```
{                                                                   \
   if( !(arg) )                                                     \
   {                                                                \
   std::ostringstream osmacro;                                      \
   osmacro << "Assert: In " __FILE__ ", line " << __LINE__          \
           << ", function " << GDCM_FUNCTION
      \
           << "\n\n";                                               \
   std::ostream &_os = gdcm::Trace::GetStream();
                \
   _os << osmacro.str() << std::endl;                                \
   assert ( arg );                                                  \
   }                                                                \
}
```

Assert.

Parameters

arg	argument to test An easy solution to pass also a message is to do: gdcmAssertMacro("my message" && 2 < 3)

Referenced by gdcm::PixelFormat::SetSamplesPerPixel().

26.235.1.4 #define gdcmDebugMacro(*msg*)

Value:

```
{                                                                   \
   if( gdcm::Trace::GetDebugFlag() )
                  \
   {                                                                \
   std::ostringstream osmacro;                                      \
   osmacro << "Debug: In " __FILE__ ", line " << __LINE__           \
           << ", function " << GDCM_FUNCTION << '\n'
      \
           << "Last system error was: "                             \
           << gdcm::System::GetLastSystemError(
      ) << '\n' << msg;  \
   std::ostream &_os = gdcm::Trace::GetStream();
                \
   _os << osmacro.str() << "\n\n" << std::endl;                      \
   }                                                                \
}
```

Debug.

Parameters

msg	message part

Referenced by gdcm::ByteValue::ByteValue(), gdcm::SequenceOfFragments::Read(), gdcm::SequenceOfItems::-Read(), gdcm::Item::Read(), gdcm::VR::Read(), and gdcm::ByteValue::SetLength().

26.235.1.5 #define gdcmErrorMacro(*msg*)

Value:

```
{                                                                   \
   if( gdcm::Trace::GetErrorFlag() )
                  \
   {                                                                \
   std::ostringstream osmacro;                                      \
   osmacro << "Error: In " __FILE__ ", line " << __LINE__           \
```

```
          << ", function " << GDCM_FUNCTION << '\n'
     \
          << msg << "\n\n";                                        \
   std::ostream &_os = gdcm::Trace::GetStream();
                \
   _os << osmacro.str() << std::endl;                               \
   }                                                                \
}
```

Error this is pretty bad, more than just warning It could mean lost of data, something not handle...

Parameters

msg	second message part

Referenced by gdcm::CommandDataSet::Insert(), gdcm::FileMetaInformation::Insert(), gdcm::DataSet::Insert(), and gdcm::Item::Read().

26.235.1.6 #define gdcmWarningMacro(*msg*)

Value:

```
{                                                                   \
   if( gdcm::Trace::GetWarningFlag() )
                  \
   {                                                                \
   std::ostringstream osmacro;                                      \
   osmacro << "Warning: In " __FILE__ ", line " << __LINE__         \
          << ", function " << GDCM_FUNCTION << "\n"
     \
          << msg << "\n\n";                                        \
   std::ostream &_os = gdcm::Trace::GetStream();
                \
   _os << osmacro.str() << std::endl;                               \
   }                                                                \
}
```

Warning.

Parameters

msg	message part

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::SequenceOfFragments::Read(), gdcm::SequenceOfItems::Read(), gdcm::Item::Read(), gdcm::Fragment::ReadValue(), and gdcm::Item::Write().

26.236 gdcmTransferSyntax.h File Reference

Classes

- class gdcm::TransferSyntax

  Class to manipulate Transfer Syntax.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)

26.237 gdcmTransferSyntaxSub.h File Reference

Classes

- class gdcm::network::TransferSyntaxSub

  TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.238 gdcmType.h File Reference

Classes

- class gdcm::Type

  Type.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)

26.239 gdcmTypes.h File Reference

Macros

- #define UINT32_MAX (4294967295U)

26.239.1 Macro Definition Documentation

26.239.1.1 #define UINT32_MAX (4294967295U)

26.240 gdcmUIDGenerator.h File Reference

Classes

- class gdcm::UIDGenerator

  Class for generating unique UID.

Namespaces

- namespace gdcm

26.241 gdcmUIDs.h File Reference

Classes

- class gdcm::UIDs

  all known uids

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)

26.242 gdcmULAction.h File Reference

Classes

- class gdcm::network::ULAction

  ULAction A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.243 gdcmULActionAA.h File Reference

Classes

- class gdcm::network::ULActionAA1
- class gdcm::network::ULActionAA2
- class gdcm::network::ULActionAA3
- class gdcm::network::ULActionAA4

- class gdcm::network::ULActionAA5
- class gdcm::network::ULActionAA6
- class gdcm::network::ULActionAA7
- class gdcm::network::ULActionAA8

Namespaces

- namespace gdcm
- namespace gdcm::network

26.244 gdcmULActionAE.h File Reference

Classes

- class gdcm::network::ULActionAE1
- class gdcm::network::ULActionAE2
- class gdcm::network::ULActionAE3
- class gdcm::network::ULActionAE4
- class gdcm::network::ULActionAE5
- class gdcm::network::ULActionAE6
- class gdcm::network::ULActionAE7
- class gdcm::network::ULActionAE8

Namespaces

- namespace gdcm
- namespace gdcm::network

26.245 gdcmULActionAR.h File Reference

Classes

- class gdcm::network::ULActionAR1
- class gdcm::network::ULActionAR10
- class gdcm::network::ULActionAR2
- class gdcm::network::ULActionAR3
- class gdcm::network::ULActionAR4
- class gdcm::network::ULActionAR5
- class gdcm::network::ULActionAR6
- class gdcm::network::ULActionAR7
- class gdcm::network::ULActionAR8
- class gdcm::network::ULActionAR9

Namespaces

- namespace gdcm
- namespace gdcm::network

26.246 gdcmULActionDT.h File Reference

Classes

- class gdcm::network::ULActionDT1
- class gdcm::network::ULActionDT2

Namespaces

- namespace gdcm
- namespace gdcm::network

26.247 gdcmULBasicCallback.h File Reference

Classes

- class gdcm::network::ULBasicCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

26.248 gdcmULConnection.h File Reference

Classes

- class gdcm::network::ULConnection

  ULConnection This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.249 gdcmULConnectionCallback.h File Reference

Classes

- class gdcm::network::ULConnectionCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

26.250 gdcmULConnectionInfo.h File Reference

Classes

- class gdcm::network::ULConnectionInfo

  ULConnectionInfo this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.251 gdcmULConnectionManager.h File Reference

Classes

- class gdcm::network::ULConnectionManager

  ULConnectionManager The ULConnectionManager performs actions on the ULConnection given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

Namespaces

- namespace gdcm
- namespace gdcm::network

26.252 gdcmULEvent.h File Reference

Classes

- class gdcm::network::ULEvent

  ULEvent base class for network events.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.253 gdcmULTransitionTable.h File Reference

Classes

- class gdcm::network::TableRow

- struct gdcm::network::Transition
- class gdcm::network::ULTransitionTable

  ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.254 gdcmULWritingCallback.h File Reference

Classes

- class gdcm::network::ULWritingCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

26.255 gdcmUNExplicitDataElement.h File Reference

Classes

- class gdcm::UNExplicitDataElement

  Class to read/write a DataElement as UNExplicit Data Element.

Namespaces

- namespace gdcm

26.256 gdcmUNExplicitImplicitDataElement.h File Reference

Classes

- class gdcm::UNExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:

Namespaces

- namespace gdcm

26.257 gdcmUnpacker12Bits.h File Reference

Classes

- class gdcm::Unpacker12Bits

  Pack/Unpack 12 bits pixel into 16bits.

Namespaces

- namespace gdcm

26.258 gdcmUsage.h File Reference

Classes

- class gdcm::Usage

  Usage.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)

26.259 gdcmUserInformation.h File Reference

Classes

- class gdcm::network::UserInformation

  UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.260 gdcmValidate.h File Reference

Classes

- class gdcm::Validate

  Validate class.

Namespaces

- namespace gdcm

26.261 gdcmValue.h File Reference

Classes

- class gdcm::Value

  Class to represent the value of a Data Element.

Namespaces

- namespace gdcm

26.262 gdcmValueIO.h File Reference

Classes

- class gdcm::ValueIO< TDE, TSwap, TType >

  Class to dispatch template calls.

Namespaces

- namespace gdcm

26.263 gdcmVersion.h File Reference

Classes

- class gdcm::Version

  major/minor and build version

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)

26.264 gdcmviewer.man File Reference

26.265 gdcmVL.h File Reference

Classes

- class gdcm::VL

  Value Length.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)

26.266 gdcmVM.h File Reference

Classes

- class gdcm::VM

  Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

Namespaces

- namespace gdcm

Macros

- #define TYPETOLENGTH(type, length)

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const VM &_val)

26.266.1 Macro Definition Documentation

26.266.1.1 #define TYPETOLENGTH(*type, length*)

Value:

```
template<> struct VMToLength<VM::type> \
  { enum { Length = length }; };
```

26.267 gdcmVR.h File Reference

Classes

- struct gdcm::UI
- class gdcm::VR

  VR class This is adapted from DICOM standard The biggest difference is the INVALID VR and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict.

Namespaces

- namespace gdcm

Macros

- #define TYPETOENCODING(type, rep, rtype)
- #define VRTypeTemplateCase(type)

Typedefs

- typedef String<'\\', 16 > gdcm::AEComp
- typedef String<'\\', 64 > gdcm::ASComp
- typedef String<'\\', 16 > gdcm::CSComp
- typedef String<'\\', 64 > gdcm::DAComp
- typedef String<'\\', 64 > gdcm::DTComp
- typedef String<'\\', 64 > gdcm::LOComp
- typedef String<'\\', 64 > gdcm::LTComp
- typedef String<'\\', 64 > gdcm::PNComp
- typedef String<'\\', 64 > gdcm::SHComp
- typedef String<'\\', 64 > gdcm::STComp
- typedef String<'\\', 16 > gdcm::TMComp
- typedef String<'\\', 64, 0 > gdcm::UIComp
- typedef String<'\\', 64 > gdcm::UTComp

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const VR &val)
- std::ostream & gdcm::operator<< (std::ostream &_os, const UI &_val)
- gdcm::TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN

Variables

- gdcm::VRBINARY

26.267.1 Macro Definition Documentation

26.267.1.1 #define TYPETOENCODING(*type, rep, rtype*)

Value:

```
template<> struct VRToEncoding<VR::type>    \
  { enum { Mode = VR::rep }; };                 \
  template<> struct VRToType<VR::type>        \
  { typedef rtype Type; };
```

26.267.1.2 #define VRTypeTemplateCase(*type*)

Value:

```
case VR::type: \
    return sizeof ( VRToType<VR::type>::Type );
```

Referenced by gdcm::VR::GetSize().

26.268 gdcmVR16ExplicitDataElement.h File Reference

Classes

- class gdcm::VR16ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.

Namespaces

- namespace gdcm

26.269 gdcmWaveform.h File Reference

Classes

- class gdcm::Waveform

  Waveform class.

Namespaces

- namespace gdcm

26.270 gdcmWin32.h File Reference

Macros

- #define GDCM_EXPORT

26.270.1 Macro Definition Documentation

26.270.1.1 #define GDCM_EXPORT

26.271 gdcmWriter.h File Reference

Classes

- class gdcm::Writer

  Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.

Namespaces

- namespace gdcm

26.272 gdcmXMLDictReader.h File Reference

Classes

- class gdcm::XMLDictReader

  Class for representing a XMLDictReader.

Namespaces

- namespace gdcm

26.273 gdcmXMLPrivateDictReader.h File Reference

Classes

- class gdcm::XMLPrivateDictReader

  Class for representing a XMLPrivateDictReader.

Namespaces

- namespace gdcm

26.274 itkGDCMImageIO2.h File Reference

Classes

- class itk::GDCMImageIO2

  ImageIO class for reading and writing DICOM V3.0 and ACR/NEMA (V1.0 & V2.0) images This class is only an adaptor to the gdcm library (currently gdcm 2.0 is used):

Namespaces

- namespace itk

Macros

- #define ITK_GDCM_EXPORT

26.274.1 Macro Definition Documentation

26.274.1.1 #define ITK_GDCM_EXPORT

26.275 README.txt File Reference

26.276 TestsList.txt File Reference

26.277 vtkGDCMImageReader.h File Reference

Classes

- class vtkGDCMImageReader

Namespaces

- namespace gdcm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

26.277.1 Macro Definition Documentation

26.277.1.1 #define VTK_CMYK 8

26.277.1.2 #define VTK_INVERSE_LUMINANCE 5

26.277.1.3 #define VTK_LOOKUP_TABLE 6

26.277.1.4 #define VTK_YBR 7

26.278 vtkGDCMImageWriter.h File Reference

Classes

- class vtkGDCMImageWriter

26.279 vtkGDCMMedicalImageProperties.h File Reference

Classes

- class vtkGDCMMedicalImageProperties

Namespaces

- namespace gdcm

26.280 vtkGDCMPolyDataReader.h File Reference

Classes

- class vtkGDCMPolyDataReader

Namespaces

- namespace gdcm

26.281 vtkGDCMPolyDataWriter.h File Reference

Classes

- class vtkGDCMPolyDataWriter

Namespaces

- namespace gdcm

26.282 vtkGDCMTesting.h File Reference

Classes

- class vtkGDCMTesting

26.283 vtkGDCMThreadedImageReader.h File Reference

Classes

- class vtkGDCMThreadedImageReader

26.284 vtkGDCMThreadedImageReader2.h File Reference

Classes

- class vtkGDCMThreadedImageReader2

26.285 vtkImageColorViewer.h File Reference

Classes

- class vtkImageColorViewer

26.286 vtkImageMapToColors16.h File Reference

Classes

- class vtkImageMapToColors16

26.287 vtkImageMapToWindowLevelColors2.h File Reference

Classes

- class vtkImageMapToWindowLevelColors2

26.288 vtkImagePlanarComponentsToComponents.h File Reference

Classes

- class vtkImagePlanarComponentsToComponents

26.289 vtkImageRGBToYBR.h File Reference

Classes

- class vtkImageRGBToYBR

26.290 vtkImageYBRToRGB.h File Reference

Classes

- class vtkImageYBRToRGB

26.291 vtkLookupTable16.h File Reference

Classes

- class vtkLookupTable16

26.292 vtkRTStructSetProperties.h File Reference

Classes

- class vtkRTStructSetProperties

Chapter 27

Example Documentation

27.1 AWTMedical3.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
package examples;

import vtk.*;
//import gdcm.*;

import vtk.util.VtkPanelContainer;
import vtk.util.VtkPanelUtil;
import vtk.util.VtkUtil;

import java.util.ArrayList;


import javax.swing.*;
import java.awt.*;
import java.io.File;

public class AWTMedical3  extends JComponent implements VtkPanelContainer {

  private vtkPanel renWin;

    vtkImageData ReadDataFile(File inSelectedFile){

        vtkImageData outImageData = null;
        Directory theDir = new Directory();

        String theInputDirectory = inSelectedFile.getPath();
        theDir.Load(theInputDirectory);

        Scanner theScanner = new Scanner();
        Tag theStudyTag = new Tag(0x0020,0x000d);
        Tag theSeriesTag = new Tag(0x0020,0x000e);
        theScanner.AddTag(theStudyTag);//get studies,
        theScanner.AddTag(theSeriesTag);//get studies,
        theScanner.Scan(theDir.GetFilenames());

        FilenamesType theStudyValues = theScanner.GetOrderedValues(theStudyTag)
      ;
        long theNumStudies = theStudyValues.size();
        //for now, take the first study, and nothing else.
        //and the return is actually not FilenamesType, just a
```

```
      //vector of strings
      if (theNumStudies != 1)
          return outImageData;
      String theStudyVal = theStudyValues.get(0);
      //now, get all the values from the scanner that are in that
      //study, then from that get their different series
      FilenamesType theFilenames =
              theScanner.GetAllFilenamesFromTagToValue(theStudyTag,
    theStudyVal);

      //from that set of filenames, isolate individual series
      //conclude that singleton series = RT struct (can do further
      //checking for things like MIPs and the like)
      //and multiple series entries = volumetric data
      theScanner.Scan(theFilenames);
      FilenamesType theSeriesValues = theScanner.GetOrderedValues(
    theSeriesTag);
      String studyUID = theScanner.GetValue(theScanner.GetFilenames().get(0),
     theStudyTag);
      long theNumSeries = theSeriesValues.size();
      for (int i = 0; i < theNumSeries; i++) {
          FilenamesType theSeriesFiles =
              theScanner.GetAllFilenamesFromTagToValue(theSeriesTag,
    theSeriesValues.get(i));
          long theNumFilesInSeries = theSeriesFiles.size();
          if (theNumFilesInSeries > 1) {//assume it's CT or volumetric data
              //for now, assume a single volume
              //could have multiples, like PET and CT

              IPPSorter sorter = new IPPSorter();
              sorter.SetComputeZSpacing(true);
              sorter.SetZSpacingTolerance(0.001);
              Boolean sorted = sorter.Sort(theSeriesFiles);
              if (!sorted){
                  //need some better way to handle failures here
                  return outImageData;
              }

              FilenamesType sortedFT = sorter.GetFilenames();
              long theSize = sortedFT.size();
              vtkStringArray sa = new vtkStringArray();
              ArrayList<String> theStrings = new ArrayList<String>();

              vtkGDCMImageReader gdcmReader = new
    vtkGDCMImageReader();
              for (int j = 0; j < theSize; j++) {
                  String theFileName = sortedFT.get(j);
                  if (gdcmReader.CanReadFile(theFileName) > 0){
                      theStrings.add(theFileName);
                      sa.InsertNextValue(theFileName);
                  } else {
                      //this is a busted series
                      //need some more appropriate error here
                      return outImageData;
                  }
              }

              gdcmReader.SetFileNames(sa);

              gdcmReader.Update();

              outImageData = gdcmReader.GetOutput();//the zeroth output
     should be the image
          }
      }
      String theImageInfo = "";
      if (outImageData != null){
          theImageInfo = outImageData.Print();
      }
      return outImageData;
  }

  //this function is a rewrite of Medical3 to see if data can
  //be loaded via gdcm easily
public AWTMedical3(File inFile) {
  // Create the buttons.
  renWin = new vtkPanel();

  vtkImageData theImageData = ReadDataFile(inFile);

  // An isosurface, or contour value of 500 is known to correspond to the
```

```
    // skin of the patient. Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter skinExtractor = new vtkContourFilter();
    skinExtractor.SetInput(theImageData);
    skinExtractor.SetValue(0, 500);
    vtkPolyDataNormals skinNormals = new vtkPolyDataNormals();
    skinNormals.SetInput(skinExtractor.GetOutput());
    skinNormals.SetFeatureAngle(60.0);
//        vtkStripper skinStripper = new vtkStripper();
//        skinStripper.SetInput(skinNormals.GetOutput());
    vtkPolyDataMapper skinMapper = new vtkPolyDataMapper();
    skinMapper.SetInput(skinNormals.GetOutput());
    skinMapper.ScalarVisibilityOff();
    vtkActor skin = new vtkActor();
    skin.SetMapper(skinMapper);
    skin.GetProperty().SetDiffuseColor(1, .49, .25);
    skin.GetProperty().SetSpecular(.3);
    skin.GetProperty().SetSpecularPower(20);

    // An isosurface, or contour value of 1150 is known to correspond to the
    // skin of the patient. Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter boneExtractor = new vtkContourFilter();
    boneExtractor.SetInput(theImageData);
    boneExtractor.SetValue(0, 1150);
    vtkPolyDataNormals boneNormals = new vtkPolyDataNormals();
    boneNormals.SetInput(boneExtractor.GetOutput());
    boneNormals.SetFeatureAngle(60.0);
    vtkStripper boneStripper = new vtkStripper();
    boneStripper.SetInput(boneNormals.GetOutput());
    vtkPolyDataMapper boneMapper = new vtkPolyDataMapper();
    boneMapper.SetInput(boneStripper.GetOutput());
    boneMapper.ScalarVisibilityOff();
    vtkActor bone = new vtkActor();
    bone.SetMapper(boneMapper);
    bone.GetProperty().SetDiffuseColor(1, 1, .9412);

    // An outline provides context around the data.
    vtkOutlineFilter outlineData = new vtkOutlineFilter();
    outlineData.SetInput(theImageData);
    vtkPolyDataMapper mapOutline = new vtkPolyDataMapper();
    mapOutline.SetInput(outlineData.GetOutput());
    vtkActor outline = new vtkActor();
    outline.SetMapper(mapOutline);
    outline.GetProperty().SetColor(0, 0, 0);

    // Now we are creating three orthogonal planes passing through the
    // volume. Each plane uses a different texture map and therefore has
    // diferent coloration.

    // Start by creatin a black/white lookup table.
    vtkLookupTable bwLut = new vtkLookupTable();
    bwLut.SetTableRange(0, 2000);
    bwLut.SetSaturationRange(0, 0);
    bwLut.SetHueRange(0, 0);
    bwLut.SetValueRange(0, 1);
    bwLut.Build();

    // Now create a lookup table that consists of the full hue circle (from
    // HSV);.
    vtkLookupTable hueLut = new vtkLookupTable();
    hueLut.SetTableRange(0, 2000);
    hueLut.SetHueRange(0, 1);
    hueLut.SetSaturationRange(1, 1);
    hueLut.SetValueRange(1, 1);
    hueLut.Build();

    // Finally, create a lookup table with a single hue but having a range
    // in the saturation of the hue.
    vtkLookupTable satLut = new vtkLookupTable();
    satLut.SetTableRange(0, 2000);
    satLut.SetHueRange(.6, .6);
    satLut.SetSaturationRange(0, 1);
    satLut.SetValueRange(1, 1);
    satLut.Build();

    // Create the first of the three planes. The filter vtkImageMapToColors
```

```
// maps the data through the corresponding lookup table created above.
// The vtkImageActor is a type of vtkProp and conveniently displays an
// image on a single quadrilateral plane. It does this using texture
// mapping and as a result is quite fast. (Note: the input image has to
// be unsigned char values, which the vtkImageMapToColors produces.);
// Note also that by specifying the DisplayExtent, the pipeline
// requests data of this extent and the vtkImageMapToColors only
// processes a slice of data.
vtkImageMapToColors saggitalColors = new vtkImageMapToColors();
saggitalColors.SetInput(theImageData);
saggitalColors.SetLookupTable(bwLut);
vtkImageActor saggital = new vtkImageActor();
saggital.SetInput(saggitalColors.GetOutput());
saggital.SetDisplayExtent(32, 32, 0, 63, 0, 92);

// Create the second (axial); plane of the three planes. We use the same
// approach as before except that the extent differs.
vtkImageMapToColors axialColors = new vtkImageMapToColors();
axialColors.SetInput(theImageData);
axialColors.SetLookupTable(hueLut);
vtkImageActor axial = new vtkImageActor();
axial.SetInput(axialColors.GetOutput());
axial.SetDisplayExtent(0, 63, 0, 63, 46, 46);

// Create the third (coronal); plane of the three planes. We use the same
// approach as before except that the extent differs.
vtkImageMapToColors coronalColors = new vtkImageMapToColors();
coronalColors.SetInput(theImageData);
coronalColors.SetLookupTable(satLut);
vtkImageActor coronal = new vtkImageActor();
coronal.SetInput(coronalColors.GetOutput());
coronal.SetDisplayExtent(0, 63, 32, 32, 0, 92);

// It is convenient to create an initial view of the data. The FocalPoint
// and Position form a vector direction. Later on (ResetCamera() method)
// this vector is used to position the camera to look at the data in
// this direction.
vtkCamera aCamera = new vtkCamera();
aCamera.SetViewUp(0, 0, -1);
aCamera.SetPosition(0, 1, 0);
aCamera.SetFocalPoint(0, 0, 0);
aCamera.ComputeViewPlaneNormal();

// Actors are added to the renderer. An initial camera view is created.
// The Dolly() method moves the camera towards the FocalPoint,
// thereby enlarging the image.
renWin.GetRenderer().AddActor(saggital);
renWin.GetRenderer().AddActor(axial);
renWin.GetRenderer().AddActor(coronal);
renWin.GetRenderer().AddActor(outline);
renWin.GetRenderer().AddActor(skin);
renWin.GetRenderer().AddActor(bone);

// Turn off bone for this example.
bone.VisibilityOff();

// Set skin to semi-transparent.
skin.GetProperty().SetOpacity(0.5);

// An initial camera view is created.  The Dolly() method moves
// the camera towards the FocalPoint, thereby enlarging the image.
renWin.GetRenderer().SetActiveCamera(aCamera);
renWin.GetRenderer().ResetCamera();
aCamera.Dolly(1.5);

// Set a background color for the renderer and set the size of the
// render window (expressed in pixels).
renWin.GetRenderer().SetBackground(1, 1, 1);
VtkPanelUtil.setSize(renWin, 640, 480);

// Note that when camera movement occurs (as it does in the Dolly()
// method), the clipping planes often need adjusting. Clipping planes
// consist of two planes: near and far along the view direction. The
// near plane clips out objects in front of the plane the far plane
// clips out objects behind the plane. This way only what is drawn
// between the planes is actually rendered.
renWin.GetRenderer().ResetCameraClippingRange();

// Setup panel
setLayout(new BorderLayout());
add(renWin, BorderLayout.CENTER);
```

```
  }



  public vtkPanel getRenWin() {
    return renWin;
  }


  public static void main(String s[]) {
    if (s.length == 0){
      return; //need a filename here
    }
    File theFile = new File(s[0]);
    //File theFile = new
       File("/Users/mmroden/Documents/MVSDownloadDirectory/Documents/1.2.840.113704.1.111.3384.1271766367.5/");
    AWTMedical3 panel = new AWTMedical3(theFile);

    JFrame frame = new JFrame("AWTMedical3");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add("Center", panel);
    frame.pack();
    frame.setVisible(true);
  }

}
```

27.2 BasicAnonymizer.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/BasicAnonymizer.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
  public MyWatcher(Subject s):base(s,"Override String"){}
  protected override void StartFilter() {
    System.Console.WriteLine( "This is my start" );
  }
  protected override void EndFilter(){
    System.Console.WriteLine( "This is my end" );
  }
  protected override void ShowProgress(Subject caller, Event evt){
    ProgressEvent pe = ProgressEvent.Cast(evt);
    System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
  }
  protected override void ShowIteration(){
    System.Console.WriteLine( "This is my iteration" );
  }
  protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 *  http://www.swig.org/Doc1.3/Java.html#adding_downcasts
 *
 *  System.Console.WriteLine( "This is my Anonymization. Type: " +
       evt.GetEventName() );
 *  System.Type type = evt.GetType();
 *  System.Console.WriteLine( "This is my Anonymization. System.Type: " +
       type.ToString() );
```

```
 *  System.Console.WriteLine( "This is my Anonymization. CheckEvent: " +
       ae.CheckEvent( evt ) );
 *  System.Console.WriteLine( "This is my Anonymization. Processing Tag #" +
       ae.GetTag().toString() );
 */
    AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
    if( ae != null )
      {
      Tag t = ae.GetTag();
      System.Console.WriteLine( "This is my Anonymization. Processing Tag #" +
      t.toString() );
      }
    else
      {
      System.Console.WriteLine( "This is my Anonymization. Unhandled Event
       type: " + evt.GetEventName() );
      }
  }
  protected override void ShowAbort(){
    System.Console.WriteLine( "This is my abort" );
  }
}

public class BasicAnonymizer
{
  public static int Main(string[] args)
    {
    gdcm.Global global = gdcm.Global.GetInstance();
    if( !global.LoadResourcesFiles() )
      {
      System.Console.WriteLine( "Could not LoadResourcesFiles" );
      return 1;
      }

    string file1 = args[0];
    string file2 = args[1];
    Reader reader = new Reader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    string certpath = gdcm.Filename.Join(gdcm.Testing.GetSourceDirectory(), "
      /Testing/Source/Data/certificate.pem" );
    gdcm.CryptographicMessageSyntax cms = new gdcm.CryptographicMessageSyntax()
      ;
    if( !cms.ParseCertificateFile( certpath ) )
      {
      return 1;
      }

    //Anonymizer ano = new Anonymizer();
    SmartPtrAno sano = Anonymizer.New();
    Anonymizer ano = sano.__ref__();

    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano,
       "Anonymizer");
    MyWatcher watcher = new MyWatcher(ano);

    ano.SetFile( reader.GetFile() );
    ano.SetCryptographicMessageSyntax( cms );
    if( !ano.BasicApplicationLevelConfidentialityProfile() )
      {
      return 1;
      }

    Writer writer = new Writer();
    writer.SetFileName( file2 );
    writer.SetFile( ano.GetFile() );
    ret = writer.Write();
    if( !ret )
      {
      return 1;
      }

    return 0;
    }
}
```

27.3 CastConvertPhilips.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################

"""
Usage:

 python --public /path/to/directory/
or
 python --private /path/to/directory/

 python --public --extension bak /path/to/directory/

rename -f 's/\.bak$//' *.bak

TODO:
http://docs.python.org/library/optparse.html#module-optparse
"""

import vtkgdcm
import vtk
import sys
import gdcm

def ProcessOneFilePublic(filename, outfilename, tmpfile):
  gdcm.ImageHelper.SetForceRescaleInterceptSlope
      (True)
  vtkreader = vtkgdcm.vtkGDCMImageReader()
  vtkreader.SetFileName( filename )
  vtkreader.Update()

  cast = vtk.vtkImageCast()
  cast.SetInput( vtkreader.GetOutput() )
  cast.SetOutputScalarTypeToUnsignedShort()

  # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file
      first:
  # Some operation will actually be discarded (we simply need a temp storage)
  vtkwriter = vtkgdcm.vtkGDCMImageWriter()
  vtkwriter.SetFileName( tmpfile )
  vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
  vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
  print "Format:",vtkreader.GetImageFormat()
  vtkwriter.SetImageFormat( vtkreader.GetImageFormat() )
  vtkwriter.SetInput( cast.GetOutput() )
  #vtkwriter.Update()
  vtkwriter.Write()

  # ok now rewrite the exact same file as the original (keep all info)
  # but use the Pixel Data Element from the written file
  tmpreader = gdcm.ImageReader()
  tmpreader.SetFileName( tmpfile )
  if not tmpreader.Read():
    sys.exit(1)

  reader = gdcm.Reader()
  reader.SetFileName( filename )
  if not reader.Read():
    sys.exit(1)

  # Make sure to remove Slope/Rescale to avoid re-execution
  ds = reader.GetFile().GetDataSet()
  tags = [
    gdcm.Tag(0x0028,0x1052),
    gdcm.Tag(0x0028,0x1053),
    gdcm.Tag(0x0028,0x1053),
  ]
  for tag in tags:
```

```
    ds.Remove( tag )

  writer = gdcm.ImageWriter()
  writer.SetFileName( outfilename )
  # Pass image from vtk written file
  writer.SetImage( tmpreader.GetImage() )
  # pass dataset from initial 'reader'
  writer.SetFile( reader.GetFile() )
  if not writer.Write():
    sys.exit(1)

def ProcessOneFilePrivate(filename, outfilename, tmpfile):
  vtkreader = vtkgdcm.vtkGDCMImageReader()
  vtkreader.SetFileName( filename )
  vtkreader.Update()


  # (2005,1409)     DS      4       0.0
  # (2005,140a)     DS      16      1.52283272283272

  # (2005,0014)     LO      26      Philips MR Imaging DD 005
  tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005"
      )
  tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005"
      )



  # Need to access some private tags, reread the file (for now):
  reader = gdcm.Reader()
  reader.SetFileName( filename )
  if not reader.Read():
    sys.exit(1)

  ds = reader.GetFile().GetDataSet()

  el1 = ds.GetDataElement( tag1 )
  el2 = ds.GetDataElement( tag2 )


  #pf = gdcm.PythonFilter()
  #pf.SetFile( reader.GetFile() )
  #print el1.GetTag()

  print el1.GetByteValue()
  v1 = eval(el1.GetByteValue().GetBuffer())
  print el2.GetByteValue()
  v2 = eval(el2.GetByteValue().GetBuffer())

  print v1
  shift = v1
  print v2
  scale = v2

  ss = vtk.vtkImageShiftScale()
  ss.SetInput( vtkreader.GetOutput() )
  # because VTK image shift / scale convention is inverted from DICOM make sure
      shift is 0
  assert shift == 0
  ss.SetShift( shift )
  ss.SetScale( scale )
  ss.SetOutputScalarTypeToUnsignedShort ()
  ss.Update()

  # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file
      first:
  # Some operation will actually be discarded (we simply need a temp storage)
  vtkwriter = vtkgdcm.vtkGDCMImageWriter()
  vtkwriter.SetFileName( tmpfile )
  vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
  vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
  vtkwriter.SetImageFormat( reader.GetImageFormat() )
  # do not pass shift/scale again
  vtkwriter.SetInput( ss.GetOutput() )
  #vtkwriter.Update()
  vtkwriter.Write()

  # ok now rewrite the exact same file as the original (keep all info)
  # but use the Pixel Data Element from the written file
  tmpreader = gdcm.ImageReader()
  tmpreader.SetFileName( tmpfile )
```

```
  if not tmpreader.Read():
    sys.exit(1)

  writer = gdcm.ImageWriter()
  writer.SetFileName( outfilename )
  # Pass image from vtk written file
  writer.SetImage( tmpreader.GetImage() )
  # pass dataset from initial 'reader'
  writer.SetFile( reader.GetFile() )
  if not writer.Write():
    sys.exit(1)

if __name__ == "__main__":

  gdcm.Trace.DebugOff()
  gdcm.Trace.WarningOff()
  #filename = sys.argv[1]
  #outfilename = sys.argv[2]
  tmpfile = "/tmp/philips_rescaled.dcm"
  #ProcessOneFile( filename, outfilename, tmpfile )
  rescaletype = sys.argv[1]
  assert rescaletype == "--public" or rescaletype == "--private"
  dirname = sys.argv[2]
  d = gdcm.Directory()
  d.Load( dirname )

  for f in d.GetFilenames():
    #print f
    ProcessOneFilePublic( f, f + ".bak", tmpfile )


print "success"
```

27.4 ChangeSequenceUltrasound.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmSmartPointer.h"
#include "gdcmDataSetHelper.h"

/*
 ./ChangeSequenceUltrasound gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

 This is the exact C++ translation of the original python example:
       ManipulateSequence.py
 */

int main(int argc, char* argv[] )
{
  if( argc < 0 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if (! reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
```

```
  gdcm::DataSet &ds = file.GetDataSet();
  gdcm::Tag tsis(0x0008,0x2112); // SourceImageSequence
  if ( ds.FindDataElement( tsis ) )
    {
    const gdcm::DataElement &sis = ds.GetDataElement
      ( tsis );
    gdcm::SmartPointer<gdcm::SequenceOfItems>
       sqsis = sis.GetValueAsSQ();
    if ( sqsis && sqsis->GetNumberOfItems() )
      {
      gdcm::Item &item1 = sqsis->GetItem(1);
      gdcm::DataSet &nestedds = item1.GetNestedDataSet
      ();
      gdcm::Tag tprcs(0x0040,0xa170); //
       PurposeOfReferenceCodeSequence
      if( nestedds.FindDataElement( tprcs ) )
        {
        const gdcm::DataElement &prcs = nestedds.
      GetDataElement( tprcs );
        gdcm::SmartPointer<gdcm::SequenceOfItems>
       sqprcs = prcs.GetValueAsSQ();
        if ( sqprcs && sqprcs->GetNumberOfItems() )
          {
          gdcm::Item &item2 = sqprcs->GetItem(1);
          gdcm::DataSet &nestedds2 = item2.GetNestedDataSet
      ();
          // (0008,0104) LO [Uncompressed predecessor]               #  24, 1
       CodeMeaning
          gdcm::Tag tcm(0x0008,0x0104);
          if( nestedds2.FindDataElement( tcm ) )
            {
            gdcm::DataElement cm = nestedds2.GetDataElement
      ( tcm );
            std::string mystr = "GDCM was here";
            cm.SetByteValue( mystr.c_str(), mystr.size() );
            nestedds2.Replace( cm );
            }
          }
        }
      }
    }

  gdcm::Writer writer;
  writer.SetFile( file );
  writer.SetFileName( outfilename );
  if ( !writer.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.5 CheckBigEndianBug.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * WARNING: This is a dev tool, do not use !
 *
 * Usage: after a gdcmconv, you would like to know if the conversion process is
       acceptable
 * sometime a vbindiff is acceptable, sometime it is not. In the case of the
       famous Philips
 * Little/Big Endian Explicit Transfer Syntax it is not easy to compare two
       files. However
```

```
 * this only impact byte ordering, thus we can compute byte-indenpendant
       information to still
 * compare the files.
 */

#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmSystem.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input1.dcm input2.dcm" << std::endl;
    return 1;
    }
  const char *filename1 = argv[1];
  const char *filename2 = argv[2];

  gdcm::ImageReader reader1;
  reader1.SetFileName( filename1 );
  if( !reader1.Read() )
    {
    std::cerr << "Could not read: " << filename1 << std::endl;
    return 1;
    }

  gdcm::ImageReader reader2;
  reader2.SetFileName( filename2 );
  if( !reader2.Read() )
    {
    std::cerr << "Could not read: " << filename2 << std::endl;
    return 1;
    }

  // TODO: need a DataSet== operator implementation

  std::cout << "Both files can be read and looks like DICOM" << std::endl;

  size_t s1 = gdcm::System::FileSize(filename1);
  size_t s2 = gdcm::System::FileSize(filename2);

  if( s1 != s2 )
    {
    std::cout << "Size mismatch: " << s1 << " != " << s2 << std::endl;
    return 1;
    }
  else
    {
    std::cout << "Size match: " << s1 << " = " << s2 << std::endl;
    }

  std::ifstream is1( filename1 );
  char *buffer1 = new char[s1];
  is1.read(buffer1, s1);

  std::ifstream is2( filename2 );
  char *buffer2 = new char[s2];
  is2.read(buffer2, s2);

  assert( s1 == s2 );
  if( memcmp(buffer1, buffer2, s1 ) == 0 )
    {
    std::cout << "memcmp succeed ! File are bit identical" << std::endl;
    }
  else
    {
    std::cout << "memcmp failed!" << std::endl;
    }

  // Hum...memcmp failed, for big endian/ little endian inversion the histogram
       of bytes
  // should still be the same. So let's compute it
  // buffer2[0] = 1; // let's make the test fail
  std::multiset<char> set1( buffer1, buffer1 + s1 );
  std::multiset<char> set2( buffer2, buffer2 + s2 );
```

```
  if( set1 == set2 )
    {
    std::cout << "set1 == set2. Byte histogram seems valid" << std::endl;
    }
  else
    {
    std::cout << "set1 != set2" << std::endl;
    }
  delete[] buffer1;
  delete[] buffer2;


  return 0;
}
```

27.6 ClinicalTrialAnnotate.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Dummy implementation of C.7.1.3 Clinical Trial Subject Module
 *
 * Usage:
 *  ClinicalTrialAnnotate gdcmData/012345.002.050.dcm out.dcm
 */

#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAnonymizer.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
    }

  // The output of gdcm::Reader is a gdcm::File
  //gdcm::File &file = reader.GetFile();

  // the dataset is the the set of element we are interested in:
  //gdcm::DataSet &ds = file.GetDataSet();

  gdcm::Anonymizer ano;
  ano.SetFile( reader.GetFile() );
  ano.RemoveGroupLength();
  ano.RemovePrivateTags();

  // PS 3.3 - 2008
  // C.7.1.3 Clinical Trial Subject Module
  // <entry group="0012" element="0010" vr="LO" vm="1" name="Clinical Trial
       Sponsor Name"/>
  ano.Replace( gdcm::Tag(0x12,0x10), "BigCompany name" );
```

```
  // <entry group="0012" element="0020" vr="LO" vm="1" name="Clinical Trial
       Protocol ID"/>
  ano.Replace( gdcm::Tag(0x12,0x20), "My Clinical Trial
       Protocol ID" );
  // <entry group="0012" element="0021" vr="LO" vm="1" name="Clinical Trial
       Protocol Name"/>
  ano.Replace( gdcm::Tag(0x12,0x21), "My Clinical Trial
       Protocol Name" );
  // <entry group="0012" element="0030" vr="LO" vm="1" name="Clinical Trial
       Site ID"/>
  ano.Replace( gdcm::Tag(0x12,0x30), "My Clinical Trial Site ID
      " );
  // <entry group="0012" element="0031" vr="LO" vm="1" name="Clinical Trial
       Site Name"/>
  ano.Replace( gdcm::Tag(0x12,0x31), "My Clinical Trial Site
       Name" );
  // <entry group="0012" element="0040" vr="LO" vm="1" name="Clinical Trial
       Subject ID"/>
  ano.Replace( gdcm::Tag(0x12,0x40), "My Clinical Trial Subject
       ID" );
  // <entry group="0012" element="0042" vr="LO" vm="1" name="Clinical Trial
       Subject Reading ID"/>
  ano.Replace( gdcm::Tag(0x12,0x42), "My Clinical Trial Subject
       Reading ID" );


  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.7 ClinicalTrialIdentificationWorkflow.cs

This is a C# example on how to use gdcm::Anonymizer

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Typical usage on UNIX:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ClinicalTrialIdentificationWorkflow.exe input_dir output_dir
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
  public MyWatcher(Subject s):base(s,"Override String"){}
  protected override void StartFilter() {
    System.Console.WriteLine( "This is my start" );
  }
  protected override void EndFilter(){
    System.Console.WriteLine( "This is my end" );
  }
  protected override void ShowProgress(Subject caller, Event evt){
    ProgressEvent pe = ProgressEvent.Cast(evt);
    System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
  }
  protected override void ShowIteration(){
```

```
    System.Console.WriteLine( "This is my iteration" );
  }
  protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 *  http://www.swig.org/Doc1.3/Java.html#adding_downcasts
 *
 *  System.Console.WriteLine( "This is my Anonymization. Type: " +
       evt.GetEventName() );
 *  System.Type type = evt.GetType();
 *  System.Console.WriteLine( "This is my Anonymization. System.Type: " +
       type.ToString() );
 *  System.Console.WriteLine( "This is my Anonymization. CheckEvent: " +
       ae.CheckEvent( evt ) );
 *  System.Console.WriteLine( "This is my Anonymization. Processing Tag #" +
       ae.GetTag().toString() );
 */
    AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
    if( ae != null )
      {
      Tag t = ae.GetTag();
      System.Console.WriteLine( "This is my Anonymization. Processing Tag #" +
      t.toString() );
      }
    else
      {
      System.Console.WriteLine( "This is my Anonymization. Unhandled Event
       type: " + evt.GetEventName() );
      }
  }
  protected override void ShowAbort(){
    System.Console.WriteLine( "This is my abort" );
  }
}

public class ClinicalTrialIdentificationWorkflow
{
  public static bool ProcessOneFile( gdcm.Anonymizer ano , string filename,
      string outfilename )
    {
    Reader reader = new Reader();
    reader.SetFileName( filename );
    bool ret = reader.Read();
    if( !ret )
      {
      return false;
      }
    // Pass in the file:
    ano.SetFile( reader.GetFile() );

    // First step, let's protect all Patient information as per
    // PS 3.15 / E.1 / Basic Application Level Confidentiality Profile
    if( !ano.BasicApplicationLevelConfidentialityProfile() )
      {
      return false;
      }

    // Now let's pass in all Clinical Trial fields
    // PS 3.3 - 2008 / C.7.1.3 Clinical Trial Subject Module
    /*
    Clinical Trial Sponsor Name (0012,0010) 1 The name of the clinical trial
       sponsor. See C.7.1.3.1.1.
    Clinical Trial Protocol ID (0012,0020) 1 Identifier for the noted protocol.
       See C.7.1.3.1.2.
    Clinical Trial Protocol Name (0012,0021) 2 The name of the clinical trial
       protocol. See C.7.1.3.1.3.
    Clinical Trial Site ID (0012,0030) 2 The identifier of the site responsible
       for submitting clinical trial data. See C.7.1.3.1.4.
    Clinical Trial Site Name (0012,0031) 2 Name of the site responsible for
       submitting clinical trial data. See C.7.1.3.1.5
    Clinical Trial Subject ID (0012,0040) 1C The assigned identifier for the
       clinical trial subject. See C.7.1.3.1.6. Shall be present if Clinical Trial
       Subject Reading ID (0012,0042) is absent. May be present otherwise.
    Clinical Trial Subject Reading ID (0012,0042) 1C Identifies the subject for
       blinded evaluations. Shall be present if Clinical Trial Subject ID (0012,0040)
       is absent.  May be present otherwise. See C.7.1.3.1.7.
     */
    ano.Replace( new gdcm.Tag(0x0012,0x0010), "MySponsorName");
    ano.Replace( new gdcm.Tag(0x0012,0x0020), "MyProtocolID");
    ano.Replace( new gdcm.Tag(0x0012,0x0021), "MyProtocolName");
    ano.Replace( new gdcm.Tag(0x0012,0x0030), "MySiteId");
```

```
  ano.Replace( new gdcm.Tag(0x0012,0x0031), "MySiteName");
  ano.Replace( new gdcm.Tag(0x0012,0x0040), "MySponsorId");
  ano.Replace( new gdcm.Tag(0x0012,0x0050), "MyTPId");
  ano.Replace( new gdcm.Tag(0x0012,0x0051), "MyTPDescription");

  // The following two are not required as they are guaranteed to be filled
     in by the
  // Basic Application Level Confidentiality Profile. Only override if you
     understand what
  // you are doing
  //ano.Replace( new gdcm.Tag(0x0012,0x0062), "YES");
  //ano.Replace( new gdcm.Tag(0x0012,0x0063), "My Super Duper Anonymization
     Overload");

  // We might be generating a subdirectory. Let's make sure the subdir exist:
  gdcm.Filename fn = new gdcm.Filename( outfilename );
  string subdir = fn.GetPath();
  if( !gdcm.PosixEmulation.MakeDirectory( subdir ) )
    {
    return false;
    }

  gdcm.FileMetaInformation fmi = ano.GetFile().GetHeader();
  // The following three lines make sure to regenerate any value:
  fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
  fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
  fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

  Writer writer = new Writer();
  writer.SetFileName( outfilename );
  writer.SetFile( ano.GetFile() );
  ret = writer.Write();
  if( !ret )
    {
    return false;
    }

  return true;
  }

public static int Main(string[] args)
  {
  gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My ClinicalTrial
     App" );

  // http://www.oid-info.com/get/1.3.6.1.4.17434
  string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
  gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
  System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot()
     );

  gdcm.Global global = gdcm.Global.GetInstance();
  if( !global.LoadResourcesFiles() )
    {
    System.Console.WriteLine( "Could not LoadResourcesFiles" );
    return 1;
    }

  if( args.Length != 2 )
    {
    System.Console.WriteLine( "Usage:" );
    System.Console.WriteLine( "ClinicalTrialIdentificationWorkflow input_dir
     output_dir" );
    return 1;
    }
  string dir1 = args[0];
  string dir2 = args[1];

  // Check input is valid:
  if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
    {
    System.Console.WriteLine( "Input directory: " + dir1 + " does not exist.
     Sorry" );
    return 1;
    }
  if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
    {
    System.Console.WriteLine( "Output directory: " + dir2 + " does not exist.
     Sorry" );
    return 1;
    }
```

```
    // Recursively search all file within this toplevel directory:
    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );
    if(nfiles == 0) return 1;

    // Let's use the pre-shipped certificate of GDCM.
    string certpath = gdcm.Filename.Join(gdcm.Testing.GetSourceDirectory(), "
      /Testing/Source/Data/certificate.pem" );
    gdcm.CryptographicMessageSyntax cms = new gdcm.CryptographicMessageSyntax()
      ;
    if( !cms.ParseCertificateFile( certpath ) )
      {
      System.Console.WriteLine( "PEM Certificate : " + certpath + " could not
       be read. Sorry" );
      return 1;
      }

    //Anonymizer ano = new Anonymizer();
    // A reference to an actual C++ instance is required here:
    SmartPtrAno sano = Anonymizer.New();
    Anonymizer ano = sano.__ref__();

    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano,
       "Anonymizer");
    MyWatcher watcher = new MyWatcher(ano);

    // Explicitely specify the Cryptographic Message Syntax to use:
    ano.SetCryptographicMessageSyntax( cms );

    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
      {
      string filename = filenames[ (int)i ];
      string outfilename = filename.Replace( dir1, dir2 );
      System.Console.WriteLine( "Filename: " + filename );
      System.Console.WriteLine( "Out Filename: " + outfilename );
      if( !ProcessOneFile( ano , filename, outfilename ) )
        {
        System.Console.WriteLine( "Could not process filename: " + filename );
        return 1;
        }
      }

    return 0;
    }
}
```

27.8 CompressImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 *
 */

#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"
#include "gdcmImageChangeTransferSyntax.h"

#include <iostream>
#include <fstream>
```

```
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
    }

  // The output of gdcm::Reader is a gdcm::File
  //gdcm::File &file = reader.GetFile();

  // the dataset is the the set of element we are interested in:
  //gdcm::DataSet &ds = file.GetDataSet();

  const gdcm::Image &image = reader.GetImage();
  image.Print( std::cout );

  gdcm::ImageChangeTransferSyntax change;
  change.SetTransferSyntax(
      gdcm::TransferSyntax::JPEG2000Lossless );
  change.SetTransferSyntax(
      gdcm::TransferSyntax::JPEGLosslessProcess14_1
       );
  //change.SetTransferSyntax( gdcm::TransferSyntax::JPEGBaselineProcess1 );
  //change.SetTransferSyntax( image.GetTransferSyntax() );
  change.SetInput( image );
  bool b = change.Change();
  if( !b )
    {
    std::cerr << "Could not change the Transfer Syntax" << std::endl;
    return 1;
    }

  //std::ofstream out( outfilename );
  //image.GetBuffer2(out);
  //out.close();
  gdcm::ImageWriter writer;
  writer.SetImage( change.GetOutput() );
  writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.9 CompressLossyJPEG.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
```

```
 * $ mono bin/CompressLossyJPEG.exe input.dcm output.dcm
 */

using System;
using gdcm;

public class CompressLossyJPEG
{
  public static int Main(string[] args)
    {
    if( args.Length < 2 )
      {
      System.Console.WriteLine( " input.dcm output.dcm" );
      return 1;
      }
    string filename = args[0];
    string outfilename = args[1];

    ImageReader reader = new ImageReader();
    reader.SetFileName( filename );
    if( !reader.Read() )
      {
      System.Console.WriteLine( "Could not read: " + filename );
      return 1;
      }

    // The output of gdcm::Reader is a gdcm::File
    File file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    DataSet ds = file.GetDataSet();

    Image image = reader.GetImage();
    //image.Print( cout );

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    TransferSyntax targetts =  new TransferSyntax( TransferSyntax.TSType.
      JPEGBaselineProcess1 );
    change.SetTransferSyntax( targetts );

    // Setup our JPEGCodec, warning it should be compatible with
       JPEGBaselineProcess1
    JPEGCodec jpegcodec = new JPEGCodec();
    if( !jpegcodec.CanCode( targetts ) )
      {
      System.Console.WriteLine( "Something went really wrong, JPEGCodec cannot
       handle JPEGBaselineProcess1" );
      return 1;
      }
    jpegcodec.SetLossless( false );
    jpegcodec.SetQuality( 50 ); // poor quality !
    change.SetUserCodec( jpegcodec ); // specify the codec to use to the
       ImageChangeTransferSyntax

    change.SetInput( image );
    bool b = change.Change();
    if( !b )
      {
      System.Console.WriteLine( "Could not change the Transfer Syntax" );
      return 1;
      }

    ImageWriter writer = new ImageWriter();
    writer.SetImage( (gdcm.Image)change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfilename );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write: " + outfilename );
      return 1;
      }

    return 0;
    }

}
```

27.10 Convert16BitsTo8Bits.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"

#include "gdcmTesting.h"
// The following file is 16/16/15 but the scalar range of the image is [0,192]
// it could be safely stored as 8bits instead:
// gdcmData/012345.002.050.dcm

int main(int, char *[])
{
  const char *directory = gdcm::Testing::GetDataRoot(
      );
  if(!directory) return 1;
  std::string file = std::string(directory) + "/012345.002.050.dcm";
  std::cout << file << std::endl;

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New
      ();
  reader->SetFileName( file.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );

  vtkImageCast *cast = vtkImageCast::New();
  cast->SetInput( reader->GetOutput() );
  cast->SetOutputScalarTypeToUnsignedChar();


  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
      ();
  writer->SetFileName( "/tmp/cast.dcm" );
  writer->SetInput( cast->GetOutput() );
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->SetMedicalImageProperties( reader->
      GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines()
       );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->Write();

  reader->Delete();
  cast->Delete();
  writer->Delete();

  return 0;
}
```

27.11 ConvertMPL.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
#     PURPOSE.  See the above copyright notice for more information.
#
##########################################################################

"""
display a DICOM image with matPlotLib via numpy

Caveats:
- Does not support UINT12/INT12

Usage:

 python ConvertNumpy.py "IM000000"

Thanks:
  plotting example - Ray Schumacher 2009
"""

import gdcm
import numpy
from pylab import *


def get_gdcm_to_numpy_typemap():
    """Returns the GDCM Pixel Format to numpy array type mapping."""
    _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.int8,
                gdcm.PixelFormat.INT8   :numpy.uint8,
                gdcm.PixelFormat.UINT16 :numpy.uint16,
                gdcm.PixelFormat.INT16  :numpy.int16,
                gdcm.PixelFormat.UINT32 :numpy.uint32,
                gdcm.PixelFormat.INT32  :numpy.int32,
                gdcm.PixelFormat.FLOAT32:numpy.float32,
                gdcm.PixelFormat.FLOAT64:numpy.float64 }
    return _gdcm_np

def get_numpy_array_type(gdcm_pixel_format):
    """Returns a numpy array typecode given a GDCM Pixel Format."""
    return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]

def gdcm_to_numpy(image):
    """Converts a GDCM image to a numpy array.
    """
    pf = image.GetPixelFormat().GetScalarType()
    print 'pf', pf
    print image.GetPixelFormat().GetScalarTypeAsString()
    assert pf in get_gdcm_to_numpy_typemap().keys(), \
           "Unsupported array type %s"%pf
    d = image.GetDimension(0), image.GetDimension(1)
    print 'Image Size: %d x %d' % (d[0], d[1])
    dtype = get_numpy_array_type(pf)
    gdcm_array = image.GetBuffer()
    ## use float for accurate scaling
    result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
    ## optional gamma scaling
    #maxV = float(result[result.argmax()])
    #result = result + .5*(maxV-result)
    #result = numpy.log(result+50) ## apprx background level
    result.shape = d
    return result

if __name__ == "__main__":
  import sys
  r = gdcm.ImageReader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read():  sys.exit(1)
  numpy_array = gdcm_to_numpy( r.GetImage() )

  subplot(111)# one plot, on left
  title(filename)
  ## many colormaps are available
  imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
  ## set the plot sizes and placement
  subplots_adjust(bottom=0.1, right=0.8, top=0.9)
  cax = axes([0.85, 0.1, 0.075, 0.8])
  colorbar(cax=cax)
  title('values')
  get_current_fig_manager().window.title('plot')
  show()
```

27.12 ConvertMultiFrameToSingleFrame.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkStringArray.h"

#include "gdcmTesting.h"
#include "gdcmFilenameGenerator.h"

int main(int argc, char *argv[])
{
  std::string filename;
  if( argc <= 1 )
    {
    const char *directory = gdcm::Testing::GetDataRoot
      ();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/US-PAL-8-10x-echo.dcm";
    filename = file;
    }
  else
    {
    filename = argv[1];
    }
  std::cout << "file: " << filename << std::endl;

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New
      ();
  reader->SetFileName( filename.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );

  int dims[3];
  reader->GetOutput()->GetDimensions( dims );

  std::ostringstream os;
  os << "singleframe";
  os << "%04d.dcm";
  gdcm::FilenameGenerator fg;
  fg.SetPattern( os.str().c_str() );
  unsigned int nfiles = dims[2];
  fg.SetNumberOfFilenames( nfiles );
  bool b = fg.Generate();
  if( !b )
    {
    std::cerr << "FilenameGenerator::Generate() failed" << std::endl;
    return 1;
    }
  if( !fg.GetNumberOfFilenames() )
    {
    std::cerr << "FilenameGenerator::Generate() failed somehow..." << std::endl
      ;
    return 1;
    }

  // By default write them as Secondary Capture (for portability)
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
      ();
  vtkStringArray *filenames = vtkStringArray::New();
  for(unsigned int i = 0; i < fg.GetNumberOfFilenames(); ++
      i)
    {
    filenames->InsertNextValue( fg.GetFilename(i) );
    }
  assert( filenames->GetNumberOfValues() == (int)fg.GetNumberOfFilenames
      () );
```

```
  writer->SetFileNames( filenames );
  filenames->Delete();
  writer->SetFileDimensionality( 2 );
  writer->SetInput( reader->GetOutput() );
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->Write();

  reader->Delete();
  writer->Delete();

  return 0;
}
```

27.13 ConvertNumpy.py

```
##########################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##########################################################################

"""
This module add support for converting a gdcm.Image to a numpy array.

Caveats:
- Does not support UINT12/INT12

Removed:
- float16 is defined in GDCM API but no implementation exist for it ...
"""

import gdcm
import numpy

def get_gdcm_to_numpy_typemap():
    """Returns the GDCM Pixel Format to numpy array type mapping."""
    _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.int8,
                gdcm.PixelFormat.INT8   :numpy.uint8,
                #gdcm.PixelFormat.UINT12 :numpy.uint12,
                #gdcm.PixelFormat.INT12  :numpy.int12,
                gdcm.PixelFormat.UINT16 :numpy.uint16,
                gdcm.PixelFormat.INT16  :numpy.int16,
                gdcm.PixelFormat.UINT32 :numpy.uint32,
                gdcm.PixelFormat.INT32  :numpy.int32,
                #gdcm.PixelFormat.FLOAT16:numpy.float16,
                gdcm.PixelFormat.FLOAT32:numpy.float32,
                gdcm.PixelFormat.FLOAT64:numpy.float64 }
    return _gdcm_np

def get_numpy_array_type(gdcm_pixel_format):
    """Returns a numpy array typecode given a GDCM Pixel Format."""
    return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]

def gdcm_to_numpy(image):
    """Converts a GDCM image to a numpy array.
    """
    pf = image.GetPixelFormat()

    assert pf.GetScalarType() in get_gdcm_to_numpy_typemap().keys(), \
           "Unsupported array type %s"%pf

    shape = image.GetDimension(0) * image.GetDimension(1),
      pf.GetSamplesPerPixel()
    if image.GetNumberOfDimensions() == 3:
      shape = shape[0] * image.GetDimension(2), shape[1]

    dtype = get_numpy_array_type(pf.GetScalarType())
    gdcm_array = image.GetBuffer()
    result = numpy.frombuffer(gdcm_array, dtype=dtype)
```

```
    result.shape = shape
    return result

if __name__ == "__main__":
  import sys
  r = gdcm.ImageReader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read():
    sys.exit(1)

  numpy_array = gdcm_to_numpy( r.GetImage() )
  print numpy_array
```

27.14 ConvertPIL.py

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

"""
save a DICOM image with PIL via numpy

Caveats:
- Does not support UINT12/INT12

Usage:

 python ConvertNumpy.py "IM000000"

Thanks:
  plotting example - Ray Schumacher 2009
"""

import gdcm
import numpy
from PIL import Image, ImageOps


def get_gdcm_to_numpy_typemap():
    """Returns the GDCM Pixel Format to numpy array type mapping."""
    _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.int8,
                gdcm.PixelFormat.INT8   :numpy.uint8,
                gdcm.PixelFormat.UINT16 :numpy.uint16,
                gdcm.PixelFormat.INT16  :numpy.int16,
                gdcm.PixelFormat.UINT32 :numpy.uint32,
                gdcm.PixelFormat.INT32  :numpy.int32,
                gdcm.PixelFormat.FLOAT32:numpy.float32,
                gdcm.PixelFormat.FLOAT64:numpy.float64 }
    return _gdcm_np

def get_numpy_array_type(gdcm_pixel_format):
    """Returns a numpy array typecode given a GDCM Pixel Format."""
    return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]

def gdcm_to_numpy(image):
    """Converts a GDCM image to a numpy array.
    """
    pf = image.GetPixelFormat().GetScalarType()
    print 'pf', pf
    print image.GetPixelFormat().GetScalarTypeAsString()
    assert pf in get_gdcm_to_numpy_typemap().keys(), \
           "Unsupported array type %s"%pf
    d = image.GetDimension(0), image.GetDimension(1)
    print 'Image Size: %d x %d' % (d[0], d[1])
    dtype = get_numpy_array_type(pf)
    gdcm_array = image.GetBuffer()
```

```
    result = numpy.frombuffer(gdcm_array, dtype=dtype)
    maxV = float(result[result.argmax()])
    ## linear gamma adjust
    #result = result + .5*(maxV-result)
    ## log gamma
    result = numpy.log(result+50) ## 50 is apprx background level
    maxV = float(result[result.argmax()])
    result = result*(2.**8/maxV) ## histogram stretch
    result.shape = d
    return result

if __name__ == "__main__":
  import sys
  r = gdcm.ImageReader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read():  sys.exit(1)
  numpy_array = gdcm_to_numpy( r.GetImage() )
  ## L is 8 bit grey
  ## http://www.pythonware.com/library/pil/handbook/concepts.htm
  pilImage = Image.frombuffer('L',
                          numpy_array.shape,
                          numpy_array.astype(numpy.uint8),
                          'raw','L',0,1)
  ## cutoff removes background noise and spikes
  pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
  pilImage.save(sys.argv[1]+'.jpg')
```

27.15 ConvertRGBToLuminance.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageLuminance.h"

#include "gdcmTesting.h"

// There is no such thing as MR Image Storage + Photometric Interpretation =
      RGB
// let's rewrite that into a proper single component image:
int main(int, char *[])
{
  const char *directory = gdcm::Testing::GetDataRoot(
      );
  if(!directory) return 1;
  std::string file = std::string(directory) + "/SIEMENS-MR-RGB-16Bits.dcm";
  std::cout << file << std::endl;

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New
      ();
  reader->SetFileName( file.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );

  vtkImageLuminance *luminance = vtkImageLuminance::New();
  luminance->SetInput( reader->GetOutput() );


  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
      ();
  writer->SetFileName( "/tmp/bla.dcm" );
  writer->SetInput( luminance->GetOutput() );
  //writer->SetImageFormat( reader->GetImageFormat() ); // Do NOT pass image
      format
```

```
  writer->SetMedicalImageProperties( reader->
      GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines()
       );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->Write();

  // TODO:
  //vtkImageAppendComponents.h

  reader->Delete();
  luminance->Delete();
  writer->Delete();

  return 0;
}
```

27.16 ConvertSingleBitTo8Bits.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"
#include "vtkPointData.h"
#include "vtkBitArray.h"
#include "vtkUnsignedCharArray.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New
      ();
  reader->SetFileName( filename );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );

  vtkDataArray* array = reader->GetOutput()->GetPointData()->GetScalars();
  vtkBitArray *barray = vtkBitArray::SafeDownCast( array );
  if( !barray ) return false;
  vtkIdType nvalues = array->GetNumberOfTuples();
  vtkUnsignedCharArray *uarray = vtkUnsignedCharArray::New();
  uarray->SetNumberOfTuples( nvalues );
  for(vtkIdType i = 0; i < nvalues; ++i)
    {
    uarray->SetValue( i, barray->GetValue(i) );
    }

  vtkImageData *copy = vtkImageData::New();
  copy->SetScalarType( VTK_UNSIGNED_CHAR );
  copy->SetExtent( reader->GetOutput()->GetExtent() );
  copy->AllocateScalars();

  //uarray->Print( std::cout );
  //copy->GetPointData()->GetScalars()->Print( std::cout );
  copy->GetPointData()->SetScalars( uarray );
  uarray->Delete();
```

```
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
      ();
  writer->SetFileName( outfilename );
  //writer->SetInput( cast->GetOutput() );
  writer->SetInput( copy );
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->SetMedicalImageProperties( reader->
      GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines()
       );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->SetFileDimensionality( reader->GetFileDimensionality( ) );
  writer->Write();

  reader->Delete();
  copy->Delete();
  writer->Delete();

  return 0;
}
```

27.17 ConvertToQImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to setup the pipeline from a gdcm::ImageReader into a
 * Qt QImage data structure.
 * It only handles 2D image.
 *
 * Ref:
 * http://doc.trolltech.com/4.5/qimage.html
 *
 * Usage:
 *  ConvertToQImage gdcmData/012345.002.050.dcm output.png

 * Thanks:
 *   Sylvain ADAM (sylvain51 hotmail com) for contributing this example
 */

#include "gdcmImageReader.h"
#include <QImage>
#include <QImageWriter>

bool ConvertToFormat_RGB888(gdcm::Image const & gimage, char *buffer
      , QImage* &imageQt)
{
  const unsigned int* dimension = gimage.GetDimensions();

  unsigned int dimX = dimension[0];
  unsigned int dimY = dimension[1];

  gimage.GetBuffer(buffer);

  // Let's start with the easy case:
  if( gimage.GetPhotometricInterpretation() ==
      gdcm::PhotometricInterpretation::RGB )
    {
    if( gimage.GetPixelFormat() != gdcm::PixelFormat::UINT8
       )
      {
      return false;
      }
    unsigned char *ubuffer = (unsigned char*)buffer;
    // QImage::Format_RGB888  13  The image is stored using a 24-bit RGB format
```

```
      (8-8-8).
    imageQt = new QImage((unsigned char *)ubuffer, dimX, dimY, 3*dimX,
      QImage::Format_RGB888);
    }
  else if( gimage.GetPhotometricInterpretation() ==
       gdcm::PhotometricInterpretation::MONOCHROME2
       )
    {
    if( gimage.GetPixelFormat() == gdcm::PixelFormat::UINT8
       )
      {
      // We need to copy each individual 8bits into R / G and B:
      unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
      unsigned char *pubuffer = ubuffer;
      for(unsigned int i = 0; i < dimX*dimY; i++)
        {
        *pubuffer++ = *buffer;
        *pubuffer++ = *buffer;
        *pubuffer++ = *buffer++;
        }

      imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
      }
    else if( gimage.GetPixelFormat() == gdcm::PixelFormat::INT16
       )
      {
      // We need to copy each individual 16bits into R / G and B (truncate
       value)
      short *buffer16 = (short*)buffer;
      unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
      unsigned char *pubuffer = ubuffer;
      for(unsigned int i = 0; i < dimX*dimY; i++)
        {
        // Scalar Range of gdcmData/012345.002.050.dcm is [0,192], we could
       simply do:
        // *pubuffer++ = *buffer16;
        // *pubuffer++ = *buffer16;
        // *pubuffer++ = *buffer16;
        // instead do it right:
        *pubuffer++ = std::min(255, (32768 + *buffer16) / 255);
        *pubuffer++ = std::min(255, (32768 + *buffer16) / 255);
        *pubuffer++ = std::min(255, (32768 + *buffer16) / 255);
        buffer16++;
        }

      imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
      }
    else
      {
      std::cerr << "Pixel Format is: " << gimage.GetPixelFormat()
       << std::endl;
      return false;
      }
    }
  else
    {
    std::cerr << "Unhandled PhotometricInterpretation: " << gimage.
      GetPhotometricInterpretation() << std::endl;
    return false;
    }

  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::ImageReader ir;
  ir.SetFileName( filename );
  if(!ir.Read())
    {
    //Read failed
    return 1;
    }
```

```
  std::cout<<"Getting image from ImageReader..."<<std::endl;

  const gdcm::Image &gimage = ir.GetImage();
  //This buffer has been declared elsewhere
  char *buffer = new char[gimage.GetBufferLength()];


  QImage *imageQt = NULL;
  if( !ConvertToFormat_RGB888( gimage, buffer, imageQt ) )
    {
    return 1;
    }


  QImageWriter writer;
  writer.setFormat("png");
  writer.setFileName( outfilename );
  if( !writer.write( *imageQt ) )
    {
    return 1;
    }

//  delete[] buffer;

  return 0;
}
```

27.18 CreateARGBImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * http://www.w3.org/Graphics/PNG/inline-alpha.html
 * alphatest.png: PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
 *
 * $ convert alphatest.png alphatest.rgba
 */

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmSystem.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.rgba output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  size_t len = gdcm::System::FileSize(filename);
  std::ifstream is(filename);

  char * buf = new char[len];
  is.read(buf, len);

  gdcm::ImageWriter writer;
  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 2 );
  unsigned int dims[3] = {};
```

```
  dims[0] = 380;
  dims[1] = 287;
  image.SetDimensions( dims );
  gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8
      ;
  pf.SetSamplesPerPixel( 4 );
  image.SetPixelFormat( pf );
  gdcm::PhotometricInterpretation pi =
      gdcm::PhotometricInterpretation::ARGB;
  image.SetPhotometricInterpretation( pi );
  image.SetTransferSyntax(
      gdcm::TransferSyntax::ExplicitVRLittleEndian
       );

  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0
      x0010) );
  pixeldata.SetByteValue( buf, len );
  image.SetDataElement( pixeldata );

  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }
  delete[] buf;


  return 0;
}
```

27.19 CreateCMYKImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * http://www.w3.org/Graphics/PNG/inline-alpha.html
 * alphatest.png: PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
 *
 * $ convert alphatest.png alphatest.cmyk
 */

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmSystem.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.cmyk output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

      size_t len = gdcm::System::FileSize(filename);
      std::ifstream is(filename);

      char * buf = new char[len];
      is.read(buf, len);

  gdcm::ImageWriter writer;
```

```
  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 2 );
  unsigned int dims[3] = {};
  dims[0] = 380;
  dims[1] = 287;
  image.SetDimensions( dims );
  gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8
      ;
  pf.SetSamplesPerPixel( 4 );
  image.SetPixelFormat( pf );
  gdcm::PhotometricInterpretation pi =
      gdcm::PhotometricInterpretation::CMYK;
  image.SetPhotometricInterpretation( pi );
  image.SetTransferSyntax(
      gdcm::TransferSyntax::ExplicitVRLittleEndian
       );

      gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0
      x0010) );
      pixeldata.SetByteValue( buf, len );
      image.SetDataElement( pixeldata );

      writer.SetFileName( outfilename );
      if( !writer.Write() )
        {
        return 1;
        }
      delete[] buf;


  return 0;
}
```

27.20 CreateJPIPDataSet.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example was created during the GSOC 2011 project for
 * JPIP
 */
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr << argv[0] << " output.dcm" << std::endl;
    return 1;
    }
  const char *outfilename = argv[1];

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );

  file.GetHeader().SetDataSetTransferSyntax(
      gdcm::TransferSyntax::JPIPReferenced );
```

```
  gdcm::Anonymizer anon;
  anon.SetFile( file );

  gdcm::MediaStorage ms =
      gdcm::MediaStorage::SecondaryCaptureImageStorage
      ;

  gdcm::UIDGenerator gen;
  anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString()
       );
  std::cout << ms.GetString() << std::endl;
  anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate()
      );
  //
  anon.Replace( gdcm::Tag(0x0010,0x10), "JPIP^EXAMPLE" );
  anon.Replace( gdcm::Tag(0x0010,0x20), "012345" );
  anon.Empty( gdcm::Tag(0x0010,0x30) );
  anon.Empty( gdcm::Tag(0x0010,0x40) );
  anon.Empty( gdcm::Tag(0x0008,0x20) );
  anon.Empty( gdcm::Tag(0x0008,0x30) );
  anon.Empty( gdcm::Tag(0x0008,0x90) );
  anon.Empty( gdcm::Tag(0x0020,0x10) );
  anon.Empty( gdcm::Tag(0x0020,0x11) );
  anon.Empty( gdcm::Tag(0x0008,0x50) );
  anon.Empty( gdcm::Tag(0x0020,0x0013) );
  anon.Replace( gdcm::Tag(0x0020,0xd), gen.Generate() )
      ;
  anon.Replace( gdcm::Tag(0x0020,0xe), gen.Generate() )
      ;
  anon.Replace( gdcm::Tag(0x0008,0x64), "WSD " );

  gdcm::Attribute<0x0028,0x7FE0> at;
  at.SetValue( "http://dicom.example.com/jpipserver.cgi?target=img.jp2"
       );
  ds.Insert( at.GetAsDataElement() );

  // Need to retrieve the PixelFormat information from the given file

  if (!w.Write() )
    {
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
    }

  return 0;
}
```

27.21 CreateRAWStorage.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################

"""
    <uid value="1.2.840.10008.5.1.4.1.1.66" name="Raw Data Storage" type="SOP
       Class" part="PS 3.4" retired="false"/>
"""

import gdcm
import sys,os

if __name__ == "__main__":
  r = gdcm.Reader()
  # Will require Testing...
  dataroot = gdcm.Testing.GetDataRoot()
  filename = os.path.join( dataroot, '012345.002.050.dcm' )
```

```
  r.SetFileName( filename )
  r.Read()
  f = r.GetFile()
  ds = f.GetDataSet()

  uid = "1.2.840.10008.5.1.4.1.1.66"
#  f = gdcm.File()
#  ds = f.GetDataSet()
  de = gdcm.DataElement( gdcm.Tag(0x0008,0x0016) )
  de.SetByteValue( uid, gdcm.VL(len(uid)) )
  vr = gdcm.VR( gdcm.VR.UI )
  de.SetVR( vr )
  ds.Replace( de )

  ano = gdcm.Anonymizer()
  ano.SetFile( r.GetFile() )
  ano.RemovePrivateTags()
  ano.RemoveGroupLength()
  taglist = [
  gdcm.Tag(0x0008,0x0008),
  gdcm.Tag(0x0008,0x0022),
  gdcm.Tag(0x0008,0x0032),
  gdcm.Tag(0x0008,0x2111),
  gdcm.Tag(0x0008,0x1150),
  gdcm.Tag(0x0008,0x1155),
  gdcm.Tag(0x0008,0x0100),
  gdcm.Tag(0x0008,0x0102),
  gdcm.Tag(0x0008,0x0104),
  gdcm.Tag(0x0040,0xa170),
  gdcm.Tag(0x0008,0x2112),
  gdcm.Tag(0x0008,0x0100),
  gdcm.Tag(0x0008,0x0102),
  gdcm.Tag(0x0008,0x0104),
  gdcm.Tag(0x0008,0x9215),
  gdcm.Tag(0x0018,0x0010),
  gdcm.Tag(0x0018,0x0022),
  gdcm.Tag(0x0018,0x0050),
  gdcm.Tag(0x0018,0x0060),
  gdcm.Tag(0x0018,0x0088),
  gdcm.Tag(0x0018,0x0090),
  gdcm.Tag(0x0018,0x1040),
  gdcm.Tag(0x0018,0x1100),
  gdcm.Tag(0x0018,0x1110),
  gdcm.Tag(0x0018,0x1111),
  gdcm.Tag(0x0018,0x1120),
  gdcm.Tag(0x0018,0x1130),
  gdcm.Tag(0x0018,0x1150),
  gdcm.Tag(0x0018,0x1151),
  gdcm.Tag(0x0018,0x1152),
  gdcm.Tag(0x0018,0x1160),
  gdcm.Tag(0x0018,0x1190),
  gdcm.Tag(0x0018,0x1210),
  gdcm.Tag(0x0020,0x0012),
  gdcm.Tag(0x0020,0x0032),
  gdcm.Tag(0x0020,0x0037),
  gdcm.Tag(0x0020,0x1041),
  gdcm.Tag(0x0020,0x4000),
  gdcm.Tag(0x0028,0x0002),
  gdcm.Tag(0x0028,0x0004),
  gdcm.Tag(0x0028,0x0010),
  gdcm.Tag(0x0028,0x0011),
  gdcm.Tag(0x0028,0x0030),
  gdcm.Tag(0x0028,0x0100),
  gdcm.Tag(0x0028,0x0101),
  gdcm.Tag(0x0028,0x0102),
  gdcm.Tag(0x0028,0x0103),
  gdcm.Tag(0x0028,0x1052),
  gdcm.Tag(0x0028,0x1053),
  gdcm.Tag(0x0028,0x2110),
  gdcm.Tag(0x0028,0x2112),
  gdcm.Tag(0x7fe0,0x0010),
  gdcm.Tag(0x0018,0x0020),
  gdcm.Tag(0x0018,0x0021),
  gdcm.Tag(0x0018,0x0023),
  gdcm.Tag(0x0018,0x0025),
  gdcm.Tag(0x0018,0x0080),
  gdcm.Tag(0x0018,0x0081),
  gdcm.Tag(0x0018,0x0083),
  gdcm.Tag(0x0018,0x0084),
  gdcm.Tag(0x0018,0x0085),
  gdcm.Tag(0x0018,0x0086),
```

```
  gdcm.Tag(0x0018,0x0087),
  gdcm.Tag(0x0018,0x0091),
  gdcm.Tag(0x0018,0x0093),
  gdcm.Tag(0x0018,0x0094),
  gdcm.Tag(0x0018,0x0095),
  gdcm.Tag(0x0018,0x1088),
  gdcm.Tag(0x0018,0x1090),
  gdcm.Tag(0x0018,0x1094),
  gdcm.Tag(0x0018,0x1250),
  gdcm.Tag(0x0018,0x1251),
  gdcm.Tag(0x0018,0x1310),
  gdcm.Tag(0x0018,0x1312),
  gdcm.Tag(0x0018,0x1314),
  gdcm.Tag(0x0018,0x1315),
  gdcm.Tag(0x0018,0x1316),
  gdcm.Tag(0x0020,0x0110),
  gdcm.Tag(0x0028,0x0120),
  gdcm.Tag(0x0028,0x1050),
  gdcm.Tag(0x0028,0x1051)
  ]
  for tag in taglist:
    #print tag
    ano.Remove( tag )

  # special handling
  gen = gdcm.UIDGenerator()
  ano.Replace( gdcm.Tag(0x0008,0x9123), gen.Generate() )
  #ano.Empty( gdcm.Tag(0x0040,0x0555) )


#
#  uid = gen.Generate()
#  de.SetTag( gdcm.Tag(0x0008,0x0018) )
#  de.SetByteValue( uid, gdcm.VL(len(uid)) )
#  ds.Insert( de )

  # init FMI now:
  #fmi = f.GetHeader()
  #ts = gdcm.TransferSyntax()
  #print ts
  #fmi.SetDataSetTransferSyntax( ts ) # default
  #print fmi.GetDataSetTransferSyntax()
  #de.SetTag( gdcm.Tag(0x0002,0x0010) )
  #uid = "1.2.840.10008.1.2"
  #de.SetByteValue( uid, gdcm.VL(len(uid)) )
  #fmi.Insert( de )
#  f.SetHeader( r.GetFile().GetHeader() )

  writer = gdcm.Writer()
  writer.SetFile( ano.GetFile() )
  writer.SetFileName( "rawstorage.dcm" );
  writer.Write()
```

27.22 csa2img.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * I do not know what the format is, just guessing from info found on the net:
 *
 * http://atonal.ucdavis.edu/matlab/fmri/spm5/spm_dicom_convert.m
 *
 * This example is an attempt at understanding the format used by SIEMENS
 * their "SIEMENS CSA NON-IMAGE" DICOM file (1.3.12.2.1107.5.9.1)
 *
 * Everything done in this code is for the sole purpose of writing
```

```
      interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 */
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  // gdcmDataExtra/gdcmNonImageData/exCSA_Non-Image_Storage.dcm
  //
      PHANTOM.MR.CARDIO_COEUR_S_QUENCE_DE_REP_RAGE.9.257.2008.03.20.14.53.25.578125.43151705.IMA
  const char *filename = argv[1];

  gdcm::Reader reader; // Do not use ImageReader
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
    }

  gdcm::CSAHeader csa;
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet
      ();

  const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag
      ();
  //std::cout << t1 << std::endl;
  //const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();

  if( ds.FindDataElement( t1 ) )
    {
    csa.LoadFromDataElement( ds.GetDataElement
      ( t1 ) );
    csa.Print( std::cout );
    }
  int dims[2] = {};
  if( csa.FindCSAElementByName( "Columns" ) )
    {
    const gdcm::CSAElement &csael = csa.GetCSAElementByName
      ( "Columns" );
    std::cout << csael << std::endl;
    //const gdcm::ByteValue *bv = csael.GetByteValue();
    gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1>
       el;
    el.Set( csael.GetValue() );
    dims[0] = el.GetValue();
    std::cout << "Columns:" << el.GetValue() << std::endl;
    }

  if( csa.FindCSAElementByName( "Rows" ) )
    {
    const gdcm::CSAElement &csael2 = csa.GetCSAElementByName
      ( "Rows" );
    std::cout << csael2 << std::endl;
    gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1>
       el2;
    el2.Set( csael2.GetValue() );
    dims[1] = el2.GetValue();
    std::cout << "Rows:" << el2.GetValue() << std::endl;
    }

  double spacing[2] = { 1. , 1. };
  bool spacingfound = false;
  if( csa.FindCSAElementByName( "PixelSpacing" ) )
    {
    const gdcm::CSAElement &csael3 = csa.GetCSAElementByName
      ( "PixelSpacing" );
    if( !csael3.IsEmpty() )
      {
```

```
      std::cout << csael3 << std::endl;
      gdcm::Element<gdcm::VR::DS, gdcm::VM::VM2>
       el3;
      el3.Set( csael3.GetValue() );
      spacing[0] = el3.GetValue(0);
      spacing[1] = el3.GetValue(1);
      std::cout << "PixelSpacing:" << el3.GetValue() << "," << el3.
      GetValue(1) << std::endl;
      spacingfound = true;
      }
    }

  if( !spacingfound )
    {
    std::cerr << "Problem with PixelSpacing" << std::endl;
    //return 1;
    }
  if( !dims[0] || !dims[1] )
    {
    std::cerr << "Problem with dims" << std::endl;
    return 1;
    }

  gdcm::ImageWriter writer;

  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 2 ); // good default
  image.SetDimension(0, dims[0] );
  image.SetDimension(1, dims[1] );
  image.SetSpacing(0, spacing[0] );
  image.SetSpacing(1, spacing[1] );
  gdcm::PixelFormat pixeltype = gdcm::PixelFormat::INT16
      ; // bytepix = spm_type('int16','bits')/8;

  //unsigned long l = image.GetBufferLength();
  //const int p =  l / (dims[0] * dims[1]);

  //image.SetNumberOfDimensions( 3 );
  //image.SetDimension(2, p / pixeltype.GetPixelSize() );

  gdcm::PhotometricInterpretation pi;
  pi = gdcm::PhotometricInterpretation::MONOCHROME2
      ;
  //pixeltype.SetSamplesPerPixel(  );
  image.SetPhotometricInterpretation( pi );
  image.SetPixelFormat( pixeltype );
  //image.SetIntercept( inputimage.GetIntercept() );
  //image.SetSlope( inputimage.GetSlope() );

  //gdcm::DataElement pixeldata( gdcm::Tag(0x7fe1,0x1010) );
  //pixeldata.SetByteValue( &outbuf[0], outbuf.size() );
  gdcm::PrivateTag csanonimaget(0x7fe1,0x10,"SIEMENS CSA
       NON-IMAGE");
  const gdcm::DataElement &pixeldata = ds.GetDataElement
      ( csanonimaget );
  image.SetDataElement( pixeldata );

  std::string outfilename = "outcsa.dcm";
  //writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr << "could not write: " << outfilename << std::endl;
    return 1;
    }


  return 0;
}
```

27.23 CStoreQtProgress.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This small example show how one can use the virtual function
 * mechanism of the SimpleSubjectWatcher class to redirect progress
 * report to a custom Qt classes
 *
 * http://doc.qt.nokia.com/latest/qprogressdialog.html
 *
 * Usage:
 * CStoreQtProgress dicom.example.com 11112 gdcmData/
     MR_Spectroscopy_SIEMENS_OF.dcm
 *
 */

#include "gdcmServiceClassUser.h"
#include "gdcmSimpleSubjectWatcher.h"
#include "gdcmProgressEvent.h"
#include "gdcmDirectory.h"
#include "gdcmPresentationContextGenerator.h"

#include <QApplication>
#include <QProgressDialog>
#include <QVBoxLayout>

namespace gdcm {
/*
 * This class is a little more complicated than what this example demonstrate
 * This watcher is capable of handling nested progress. Since the Progress
 * grows from [0 to 1] on a per file basis and we only have one instance of a
 * watcher per association, we need some calculation to compute the global
 * (total) progress
 * In fact we simply divide the per-file progress by the number of files.
 *
 * This QtWatcher class will then update the progress bar according to the
 * progress.
 */
class MyQtWatcher : public SimpleSubjectWatcher
{
  size_t nfiles;
  double progress;
  size_t index;
  double refprogress;
  QWidget* win;
  QProgressDialog* qtprogress;
public:
  MyQtWatcher(Subject * s, const char *comment = "", QWidget *w = NULL,
      QProgressDialog* p = NULL, size_t n = 1):
    SimpleSubjectWatcher(s,comment),nfiles(n),progress(0),index(0),refprogress(
      0),win(w),qtprogress(p){}
  void ShowIteration()
    {
    index++;
    assert( index <= nfiles );
    // update refprogess (we are moving to the next file)
    refprogress = progress;
    }
  void ShowProgress(Subject *, const Event &evt)
    {
    // Retrieve the ProgressEvent:
    const ProgressEvent &pe = dynamic_cast<const ProgressEvent&>(evt);
    // compute global progress:
    progress = refprogress + (1. / nfiles ) * pe.GetProgress();
    // Print Global and local progress to stdout:
    std::cout << "Global Progress: " << progress << " per file progress " << pe
      .GetProgress() << std::endl;
    //set progress value in the QtProgress bar
    int i = progress * 100 + 0.5; // round to next int
    qtprogress->setValue(i);
    win->show();
    }
  virtual void ShowDataSet(Subject *caller, const Event &evt) {}
};
} // end namespace gdcm
```

```
int main(int argc, char *argv[])
{
  QApplication a(argc, argv);

  const char *remote = argv[1];
  int portno = atoi(argv[2]);
  const char *filename = argv[3];

  QVBoxLayout* layout = new QVBoxLayout;
  QWidget*  win = new QWidget;

  QProgressDialog* progress = new QProgressDialog("Sending data...", "Cancel",
      0, 100);
  progress->setWindowModality(Qt::WindowModal);

  layout->addWidget(progress,Qt::AlignCenter);
  win->setLayout(layout);

  gdcm::SmartPointer<gdcm::ServiceClassUser>
       scup = new gdcm::ServiceClassUser;
  gdcm::ServiceClassUser &scu = *scup;
  //gdcm::SimpleSubjectWatcher w( &scu, "TestServiceClassUser" );
  // let's use a more complicated progress reported in this example
  gdcm::MyQtWatcher w( &scu, "QtWatcher", win, progress );

  scu.SetHostname( remote );
  scu.SetPort( portno );
  scu.SetTimeout( 1000 );
  scu.SetCalledAETitle( "GDCM_STORE" );

  if( !scu.InitializeConnection() )
    {
    return 1;
    }

  gdcm::Directory::FilenamesType filenames;
  filenames.push_back( filename );

  // setup the PC(s) based on the filenames:
  gdcm::PresentationContextGenerator
      generator;
  if( !generator.GenerateFromFilenames(filenames) )
    {
    return 1;
    }

  // Setup PresentationContext(s)
  scu.SetPresentationContexts( generator.
      GetPresentationContexts() );

  // Start ASSOCIATION
  if( !scu.StartAssociation() )
    {
    return 1;
    }

  // Send C-STORE
  if( !scu.SendStore( filename ) )
    {
    return 1;
    }

  // Stop ASSOCIATION
  if( !scu.StopAssociation() )
    {
    return 1;
    }

  win->show();

  return a.exec();
}
```

27.24 DecompressImage.cs

This is a C# example on how to use gdcm::Image

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressImage.exe gdcmData/012345.002.050.dcm decompress.dcm
 */
using System;
using gdcm;

public class DecompressImage
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    string file2 = args[1];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    Image image = new Image();
    Image ir = reader.GetImage();

    image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );

    //Just for fun:
    //int dircos =  ir.GetDirectionCosines();
    //t = gdcm.Orientation.GetType(dircos);
    //int l = gdcm.Orientation.GetLabel(t);
    //System.Console.WriteLine( "Orientation label:" + l );

    // Set the dimensions,
    // 1. either one at a time
    //image.SetDimension(0, ir.GetDimension(0) );
    //image.SetDimension(1, ir.GetDimension(1) );

    // 2. the array at once
    uint[] dims = {0, 0};
    // Just for fun let's invert the dimensions:
    dims[0] = ir.GetDimension(1);
    dims[1] = ir.GetDimension(0);
    ir.SetDimensions( dims );

    PixelFormat pixeltype = ir.GetPixelFormat();
    image.SetPixelFormat( pixeltype );

    PhotometricInterpretation pi = ir.GetPhotometricInterpretation();
    image.SetPhotometricInterpretation( pi );

    DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
    byte[] str1 = new byte[ ir.GetBufferLength()];
    ir.GetBuffer( str1 );
    //System.Console.WriteLine( ir.GetBufferLength() );
    pixeldata.SetByteValue( str1, new VL( (uint)str1.Length ) );
    //image.SetDataElement( pixeldata );
    ir.SetDataElement( pixeldata );


    ImageWriter writer = new ImageWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( ir );
    ret = writer.Write();
    if( !ret )
      {
```

```
      return 1;
      }

    return 0;
    }
}
```

27.25 DecompressImage.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################

"""
Usage:

 python DecompressImage.py gdcmData/012345.002.050.dcm decompress.dcm
"""

import gdcm
import sys

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.ImageReader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  image = gdcm.Image()
  ir = r.GetImage()

  image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
  dims = ir.GetDimensions();
  print ir.GetDimension(0);
  print ir.GetDimension(1);
  print "Dims:",dims

  #  Just for fun:
  dircos =  ir.GetDirectionCosines()
  t = gdcm.Orientation.GetType(dircos)
  l = gdcm.Orientation.GetLabel(t)
  print "Orientation label:",l

  image.SetDimension(0, ir.GetDimension(0) );
  image.SetDimension(1, ir.GetDimension(1) );

  pixeltype = ir.GetPixelFormat();
  image.SetPixelFormat( pixeltype );

  pi = ir.GetPhotometricInterpretation();
  image.SetPhotometricInterpretation( pi );

  pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010)
       )
  str1 = ir.GetBuffer()
  #print ir.GetBufferLength()
  pixeldata.SetByteValue( str1, gdcm.VL( len(str1) ) )
  image.SetDataElement( pixeldata )

  w = gdcm.ImageWriter()
  w.SetFileName( file2 )
  w.SetFile( r.GetFile() )
  w.SetImage( image )
```

```
  if not w.Write():
    sys.exit(1)
```

27.26 DecompressImageMultiframe.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
$ gdcminfo ~/Desktop/angiogram-06.dcm
MediaStorage is 1.2.840.10008.5.1.4.1.1.12.1 [X-Ray Angiographic Image Storage]
TransferSyntax is 1.2.840.10008.1.2.4.50 [JPEG Baseline (Process 1): Default
       Transfer Syntax for Lossy JPEG 8 Bit Image Compression]
NumberOfDimensions: 3
Dimensions: (512,512,355)
Origin: (0,0,0)
Spacing: (1,1,40)
DirectionCosines: (1,0,0,0,1,0)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel    :1
BitsAllocated      :8
BitsStored         :8
HighBit            :7
PixelRepresentation:0
ScalarType found   :UINT8
PhotometricInterpretation: MONOCHROME2
PlanarConfiguration: 0
TransferSyntax: 1.2.840.10008.1.2.4.50
Orientation Label: AXIAL
*/

/*
 * Description:
 *
 * Assume we have a file angiogram-06.dcm as described above.
 * the following program will decompress directly from the extracted jpeg
       stream.
 *
 * First step extract the jpeg stream (but not the Basic Offset Table):
 *
 * $ gdcmraw -i angiogram-06.dcm -o /tmp/output/chris --split-frags --pattern
       %d.jpg
 *
 * Check that indeed there are 355 files, while there are 356 fragments in the
       original DICOM file, since
 * gdcmraw always skip the first fragment (Basic Offset Table).
 *
 * Now from those individual jpeg stream, recreate a fake gdcm.DataElement...
 *
 * Usage:
 *
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono ./bin/DecompressImageMultiframe.exe /tmp/output
 */
using System;
using gdcm;

public class DecompressImageMultiframe
{
  public static int Main(string[] args)
    {
    string directory = args[0];
    gdcm.Directory dir = new gdcm.Directory();
    uint nfiles = dir.Load(directory);
    //System.Console.WriteLine(dir.toString());
    gdcm.FilenamesType filenames = dir.GetFilenames();
```

```
    Image image = new Image();
    image.SetNumberOfDimensions( 3 ); // important for now
    DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

    // Create a new SequenceOfFragments C++ object, store it as a SmartPointer
       :
    SmartPtrFrag sq = SequenceOfFragments.New();

    // Yeah, the file are not garantee to be in order, please adapt...
    for(uint i = 0; i < nfiles; ++i)
      {
      System.Console.WriteLine( filenames[(int)i] );
      string file = filenames[(int)i];
      System.IO.FileStream infile =
        new System.IO.FileStream(file, System.IO.FileMode.Open, System.IO.
      FileAccess.Read);
      uint fsize = gdcm.PosixEmulation.FileSize(file);

      byte[] jstream  = new byte[fsize];
      infile.Read(jstream, 0 , jstream.Length);

      Fragment frag = new Fragment();
      frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
      sq.AddFragment( frag );
      }

    // Pass by reference:
    pixeldata.SetValue( sq.__ref__() );

    // insert:
    image.SetDataElement( pixeldata );

    // JPEG use YBR to achieve better compression ratio by default (not RGB)
    // FIXME hardcoded:
    PhotometricInterpretation pi = new PhotometricInterpretation(
      PhotometricInterpretation.PIType.MONOCHROME2 );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(1,8,8,7);
    image.SetPixelFormat( pixeltype );

    // FIXME hardcoded:
    image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.
      JPEGLosslessProcess14_1 ) );
    image.SetDimension(0, 512);
    image.SetDimension(1, 512);
    image.SetDimension(2, 355);

    // Decompress !
    byte[] decompressedData = new byte[(int)image.GetBufferLength()];
    image.GetBuffer(decompressedData);

    // Write out the decompressed bytes
    System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
      System.IO.File.Open(@"/tmp/dd.raw",
        System.IO.FileMode.Create))
      {
      System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
      writer.Write(decompressedData);
      }


    return 0;
    }
}
```

27.27 DecompressJPEGFile.cs

This is a C# example on how to use gdcm::SequenceOfFragments

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library
```

```
  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressJPEGFile.exe somejpegfile.jpg
 */
using System;
using gdcm;

public class DecompressJPEGFile
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    System.IO.FileStream infile =
      new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.
      FileAccess.Read);
    uint fsize = gdcm.PosixEmulation.FileSize(file1);

    byte[] jstream  = new byte[fsize];
    infile.Read(jstream, 0 , jstream.Length);

    Trace.DebugOn();
    Image image = new Image();
    image.SetNumberOfDimensions( 2 ); // important for now
    DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

    // DO NOT set a ByteValue here, JPEG is a particular kind of encapsulated
       syntax
    // in which can one cannot use a simple byte array for storage. Instead,
       see
    // gdcm.SequenceOfFragments
    //pixeldata.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );

    // Create a new SequenceOfFragments C++ object, store it as a SmartPointer
       :
    SmartPtrFrag sq = SequenceOfFragments.New();
    Fragment frag = new Fragment();
    frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
    // Single file => single fragment
    sq.AddFragment( frag );
    // Pass by reference:
    pixeldata.SetValue( sq.__ref__() );

    // insert:
    image.SetDataElement( pixeldata );

    // JPEG use YBR to achieve better compression ratio by default (not RGB)
    // FIXME hardcoded:
    PhotometricInterpretation pi = new PhotometricInterpretation(
      PhotometricInterpretation.PIType.YBR_FULL );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(3,8,8,7);
    image.SetPixelFormat( pixeltype );

    // FIXME hardcoded:
    image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.
      JPEGLosslessProcess14_1 ) );
    image.SetDimension(0, 692);
    image.SetDimension(1, 721);

    // Decompress !
    byte[] decompressedData = new byte[(int)image.GetBufferLength()];
    image.GetBuffer(decompressedData);

    // Write out the decompressed bytes
    System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
      System.IO.File.Open(@"/tmp/dd.raw",
        System.IO.FileMode.Create))
      {
```

```
      System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
      writer.Write(decompressedData);
      }


    return 0;
    }
}
```

27.28 DecompressPixmap.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * This example will take in a DICOM file, and tries to decompress it (actually
       write it
 * as ImplicitVRLittleEndian Transfer Syntax).
 *
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/DecompressPixmap.java -d
       .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java DecompressPixmap gdcmData/
      012345.002.050.dcm out.dcm
 */
import gdcm.*;

public class DecompressPixmap
{
  public static void main(String[] args) throws Exception
    {
    String file1 = args[0];
    String file2 = args[1];
    //PixmapReader reader = new PixmapReader();
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read: " + file1 );
      }

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.
      ImplicitVRLittleEndian) );
    change.SetInput( reader.GetPixmap() );
    if( !change.Change() )
      {
      throw new Exception("Could not change: " + file1 );
      }

    // When using a PixmapReader the following code crashes, I do not
       understand why (MM)
    // Instead hack our way in, and use an ImageReader instead of a
       PixmapReader
    //
    // Hum looks like Java Covariant Return type is not working for some reason
    //Pixmap out = ((PixmapToPixmapFilter)change).GetOutput(); // old syntax
    Pixmap out2 = (Pixmap)change.GetOutput(); // new syntax
    System.out.println( out2.toString() );

    // Set the Source Application Entity Title
    FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );
```

```
    PixmapWriter writer = new PixmapWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    //writer.SetImage( out );
    writer.SetImage( out2 );
    ret = writer.Write();
    if( !ret )
      {
      throw new Exception("Could not write: " + file2 );
      }

    }
}
```

27.29 DiffFile.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input1.dcm input2.dcm" << std::endl;
    return 1;
    }
  const char *filename1 = argv[1];
  const char *filename2 = argv[2];

  gdcm::Reader reader1;
  reader1.SetFileName( filename1 );
  if( !reader1.Read() )
    {
    return 1;
    }

  gdcm::Reader reader2;
  reader2.SetFileName( filename2 );
  if( !reader2.Read() )
    {
    return 1;
    }

  const gdcm::File &file1 = reader1.GetFile();
  const gdcm::File &file2 = reader2.GetFile();

  const gdcm::DataSet &ds1 = file1.GetDataSet();
  const gdcm::DataSet &ds2 = file2.GetDataSet();

  gdcm::DataSet::ConstIterator it1 = ds1.Begin
      ();
  gdcm::DataSet::ConstIterator it2 = ds2.Begin
      ();

  const gdcm::DataElement &de1 = *it1;
  const gdcm::DataElement &de2 = *it2;
  if( de1 == de2 )
    {
    }
  while( it1 != ds1.End() && it2 != ds2.End() && *it1 == *it2 )
    {
    ++it1;
    ++it2;
    }
```

```
  if( it1 != ds1.End() || it2 != ds2.End() )
    {
    std::cerr << "Problem with:" << std::endl;
    if( it1 != ds1.End() )
      {
      std::cerr << "ds1: " << *it1 << std::endl;
      }
    if( it2 != ds2.End() )
      {
      std::cerr << "ds2: " << *it2 << std::endl;
      }
    return 1;
    }

  return 0;
}
```

27.30 DiscriminateVolume.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmScanner.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"
#include "gdcmDirectionCosines.h"
#include <cmath>

/*
 * The following example is a basic sorted which should work in generic cases.
 * It sort files based on:
 * Study Instance UID
 *   Series Instance UID
 *     Frame of Reference UID
 *       Image Orientation (Patient)
 *         Image Position (Patient) (Sorting based on IPP + IOP)
 */

namespace gdcm {
  const Tag t1(0x0020,0x000d); // Study Instance UID
  const Tag t2(0x0020,0x000e); // Series Instance UID
  const Tag t3(0x0020,0x0052); // Frame of Reference UID
  const Tag t4(0x0020,0x0037); // Image Orientation (Patient)

class DiscriminateVolume
{
private:
  std::vector< Directory::FilenamesType > SortedFiles;
  std::vector< Directory::FilenamesType > UnsortedFiles;

  Directory::FilenamesType
      GetAllFilenamesFromTagToValue(
    Scanner const & s, Directory::FilenamesType const &
      filesubset, Tag const &t, const char *valueref)
{
  Directory::FilenamesType theReturn;
  if( valueref )
    {
    size_t len = strlen( valueref );
    Directory::FilenamesType::const_iterator file = filesubset.begin();
    for(; file != filesubset.end(); ++file)
      {
      const char *filename = file->c_str();
      const char * value = s.GetValue(filename, t);
      if( value && strncmp(value, valueref, len ) == 0 )
        {
        theReturn.push_back( filename );
```

```
        }
      }
    }
  return theReturn;
}

void ProcessAIOP(Scanner const & s, Directory::FilenamesType
       const & subset, const char *iopval)
{
  std::cout << "IOP: " << iopval << std::endl;
  IPPSorter ipp;
  ipp.SetComputeZSpacing( true );
  ipp.SetZSpacingTolerance( 1e-3 ); // ??
  bool b = ipp.Sort( subset );
  if( !b )
    {
    // If you reach here this means you need one more parameter to discriminiat
       this
    // series. Eg. T1 / T2 intertwinted. Multiple Echo (0018,0081)
    std::cerr << "Failed to sort: " << subset.begin()->c_str() << std::endl;
    for(
      Directory::FilenamesType::const_iterator file = subset.begin();
      file != subset.end(); ++file)
      {
      std::cerr << *file << std::endl;
      }
    UnsortedFiles.push_back( subset );
    return ;
    }
  ipp.Print( std::cout );
  SortedFiles.push_back( ipp.GetFilenames() );
}

void ProcessAFrameOfRef(Scanner const & s, Directory::FilenamesType
       const & subset, const char * frameuid)
{
  // In this subset of files (belonging to same series), let's find those
  // belonging to the same Frame ref UID:
  Directory::FilenamesType files =
      GetAllFilenamesFromTagToValue(
    s, subset, t3, frameuid);

  std::set< std::string > iopset;

  for(
    Directory::FilenamesType::const_iterator file = files.begin();
    file != files.end(); ++file)
    {
    //std::cout << *file << std::endl;
    const char * value = s.GetValue(file->c_str(), gdcm::t4 );
    assert( value );
    iopset.insert( value );
    }
  size_t n = iopset.size();
  if ( n == 0 )
    {
    assert( files.empty() );
    return;
    }

  std::cout << "Frame of Ref: " << frameuid << std::endl;
  if ( n == 1 )
    {
    ProcessAIOP(s, files, iopset.begin()->c_str() );
    }
  else
    {
    const char *f = files.begin()->c_str();
    std::cerr << "More than one IOP: " << f << std::endl;
    // Make sure that there is actually 'n' different IOP
    gdcm::DirectionCosines ref;
    gdcm::DirectionCosines dc;
    for(
      std::set< std::string >::const_iterator it = iopset.begin();
      it != iopset.end(); ++it )
      {
      ref.SetFromString( it->c_str() );
      for(
        Directory::FilenamesType::const_iterator file = files.begin();
        file != files.end(); ++file)
        {
```

```
        std::string value = s.GetValue(file->c_str(), gdcm::t4 );
        if( value != it->c_str() )
          {
          dc.SetFromString( value.c_str() );
          const double crossdot = ref.CrossDot(dc);
          const double eps = std::fabs( 1. - crossdot );
          if( eps < 1e-6 )
            {
            std::cerr << "Problem with IOP discrimination: " << file->c_str()
              << " " << it->c_str() << std::endl;
            return;
            }
          }
        }
      }
      // If we reach here this means there is actually 'n' different IOP
    for(
      std::set< std::string >::const_iterator it = iopset.begin();
      it != iopset.end(); ++it )
      {
      const char *iopvalue = it->c_str();
      Directory::FilenamesType iopfiles =
      GetAllFilenamesFromTagToValue(
        s, files, t4, iopvalue );
      ProcessAIOP(s, iopfiles, iopvalue );
      }
    }
}

void ProcessASeries(Scanner const & s, const char * seriesuid)
{
  std::cout << "Series: " << seriesuid << std::endl;
  // let's find all files belonging to this series:
  Directory::FilenamesType seriesfiles =
      GetAllFilenamesFromTagToValue(
    s, s.GetFilenames(), t2, seriesuid);

  gdcm::Scanner::ValuesType vt3 = s.GetValues(t3);
  for(
    gdcm::Scanner::ValuesType::const_iterator it = vt3.begin()
    ; it != vt3.end(); ++it )
    {
    ProcessAFrameOfRef(s, seriesfiles, it->c_str());
    }
}

void ProcessAStudy(Scanner const & s, const char * studyuid)
{
  std::cout << "Study: " << studyuid << std::endl;
  gdcm::Scanner::ValuesType vt2 = s.GetValues(t2);
  for(
    gdcm::Scanner::ValuesType::const_iterator it = vt2.begin()
    ; it != vt2.end(); ++it )
    {
    ProcessASeries(s, it->c_str());
    }
}
public:

void Print( std::ostream & os )
{
  os << "Sorted Files: " << std::endl;
  for(
    std::vector< Directory::FilenamesType >::const_iterator it = SortedFiles.
      begin();
    it != SortedFiles.end(); ++it )
    {
    os << "Group: " << std::endl;
    for(
      Directory::FilenamesType::const_iterator file = it->begin();
      file != it->end(); ++file)
      {
      os << *file << std::endl;
      }
    }
  os << "Unsorted Files: " << std::endl;
  for(
    std::vector< Directory::FilenamesType >::const_iterator it = UnsortedFiles.
      begin();
    it != UnsortedFiles.end(); ++it )
    {
```

```
    os << "Group: " << std::endl;
    for(
      Directory::FilenamesType::const_iterator file = it->begin();
      file != it->end(); ++file)
      {
      os << *file << std::endl;
      }
    }

}

  std::vector< Directory::FilenamesType > const & GetSortedFiles() const {
      return SortedFiles; }
  std::vector< Directory::FilenamesType > const & GetUnsortedFiles() const {
      return UnsortedFiles; }

void ProcessIntoVolume( Scanner const & s )
{
  gdcm::Scanner::ValuesType vt1 = s.GetValues(
      gdcm::t1 );
  for(
    gdcm::Scanner::ValuesType::const_iterator it = vt1.begin()
    ; it != vt1.end(); ++it )
    {
    ProcessAStudy( s, it->c_str() );
    }

}

};

} // namespace gdcm

int main(int argc, char *argv[])
{
  std::string dir1;
  if( argc < 2 )
    {
    const char *extradataroot = NULL;
#ifdef GDCM_BUILD_TESTING
    extradataroot = gdcm::Testing::GetDataExtraRoot
      ();
#endif
    if( !extradataroot )
      {
      return 1;
      }
    dir1 = extradataroot;
    dir1 += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
  else
    {
    dir1 = argv[1];
    }

  gdcm::Directory d;
  d.Load( dir1.c_str(), true ); // recursive !

  gdcm::Scanner s;
  s.AddTag( gdcm::t1 );
  s.AddTag( gdcm::t2 );
  s.AddTag( gdcm::t3 );
  s.AddTag( gdcm::t4 );
  bool b = s.Scan( d.GetFilenames() );
  if( !b )
    {
    std::cerr << "Scanner failed" << std::endl;
    return 1;
    }

  gdcm::DiscriminateVolume dv;
  dv.ProcessIntoVolume( s );
  dv.Print( std::cout );

  return 0;
}
```

27.31 DumbAnonymizer.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################

"""
This example shows how one can use the gdcm.Anonymizer in 'dumb' mode.
This class becomes really handy when one knows which particular tag to fill in.

Usage:

 python DumbAnonymizer.py gdcmData/012345.002.050.dcm out.dcm

"""

import gdcm

# http://www.oid-info.com/get/1.3.6.1.4.17434
THERALYS_ORG_ROOT = "1.3.6.1.4.17434"

tag_rules={
  # Value
  (0x0012,0x0010):("Value","MySponsorName"),
  (0x0012,0x0020):("Value","MyProtocolID"),
  (0x0012,0x0021):("Value","MyProtocolName"),
  (0x0012,0x0062):("Value","YES"),
  (0x0012,0x0063):("Value","MyDeidentificationMethod"),

  # Method
  #(0x0002,0x0003):("Method","GenerateMSOPId"),
  #(0x0008,0x1155):("Method","GenerateMSOPId"),
  (0x0008,0x0018):("Method","GenerateMSOPId"),
  (0x0010,0x0010):("Method","GetSponsorInitials"),
  (0x0010,0x0020):("Method","GetSponsorId"),
  (0x0012,0x0030):("Method","GetSiteId"),
  (0x0012,0x0031):("Method","GetSiteName"),
  (0x0012,0x0040):("Method","GetSponsorId"),
  (0x0012,0x0050):("Method","GetTPId"),
  (0x0018,0x0022):("Method","KeepIfExist"),
  (0x0018,0x1315):("Method","KeepIfExist"),
  (0x0020,0x000d):("Method","GenerateStudyId"),
  (0x0020,0x000e):("Method","GenerateSeriesId"),
  (0x0020,0x1002):("Method","GetNumberOfFrames"),
  (0x0020,0x0020):("Method","GetPatientOrientation"),
  # Other:
  (0x0012,0x0051):("Patient Field","Type Examen"),
  (0x0018,0x1250):("Sequence Field","Receive Coil"),
  (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
  (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
  (0x0018,0x0082):("Sequence Field","Invertion Time"),
}

class MyAnon:
  def __init__(self):
    self.studyuid = None
    self.seriesuid = None
    generator = gdcm.UIDGenerator()
    if not self.studyuid:
      self.studyuid = generator.Generate()
    if not self.seriesuid:
      self.seriesuid = generator.Generate()
  def GetSponsorInitials(self):
    return "dummy^foobar"
  def GenerateStudyId(self):
    return self.studyuid
  def GenerateSeriesId(self):
    return self.seriesuid
  #def GenerateMSOPId(self):
  def GenerateMSOPId(self):
```

```
    generator = gdcm.UIDGenerator()
    return generator.Generate()
  def GetSiteId(self):
    return "MySiteId"
  def GetSiteName(self):
    return "MySiteName"
  def GetSponsorId(self):
    return "MySponsorId"
  def GetTPId(self):
    return "MyTP"

if __name__ == "__main__":
  import sys
  gdcm.FileMetaInformation.SetSourceApplicationEntityTitle
      ( "DumbAnonymizer" )
  gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )

  r = gdcm.Reader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read(): sys.exit(1)

  obj = MyAnon()

  w = gdcm.Writer()
  ano = gdcm.Anonymizer()
  ano.SetFile( r.GetFile() )
  ano.RemoveGroupLength()
  for tag,rule in tag_rules.items():
    if rule[0] == 'Value':
      print tag,rule
      ano.Replace( gdcm.Tag( tag[0], tag[1] ), rule[1] )
    elif rule[0] == 'Method':
      print tag,rule
      # result = locals()[rule[1]]()
      methodname = rule[1]
      if hasattr(obj, methodname):
        _member = getattr(obj, methodname)
        result = _member()
        ano.Replace( gdcm.Tag( tag[0], tag[1] ), result )
      else:
        print "Problem with: ", methodname

  outfilename = sys.argv[2]
  w.SetFileName( outfilename )
  w.SetFile( ano.GetFile() )
  if not w.Write(): sys.exit(1)
```

27.32 DumpADAC.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * the goal of this example is to mimic the behavior of disp_img_header
 * see http://www.gmecorp-usa.com/IM/NM/GC/ADAC/SV/adactechtips/
      Released_01Q3.pdf
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>
#include <vector>
```

```
#include <string.h>
#include <assert.h>
#include <stdint.h>

struct dict
{
  uint16_t key;
  const char *name;
};

dict Array[] = {
  { 0x01, "Patient name" },
  { 0x02, "Patient ID" },
  { 0x03, "Patient sex" },
  { 0x04, "Patient age" },
  { 0x05, "Patient height" },
  { 0x06, "Patient weight" },
  { 0x07, "Exam date" },
  { 0x08, "Dose admin. time" },
  { 0x09, "Unique exam key" },
  { 0x0a, "Exam procedure" },
  { 0x0b, "Referring physician" },
  { 0x0c, "Attending physician" },
  { 0x0d, "Imaging modality" },
  { 0x0e, "Hospital ID" },
  { 0x0f, "Histogram crv file" },
  { 0x10, "Acq. start time" },
  { 0x11, "Object data type" },
  { 0x12, "Image viewid" },
  { 0x13, "Imaging device name" },
  { 0x14, "Device serial number" },
  { 0x15, "Collimator" },
  { 0x16, "Software version" },
  { 0x17, "Radiopharmaceutical #1" },
  { 0x18, "Energy window #1 center" },
  { 0x19, "Radiopharmaceutical #2" },
  { 0x1a, "Energy window #1 width" },
  { 0x1b, "Isotope imaging mode" },
  { 0x1c, "Energy window #2 center" },
  { 0x1d, "Energy window #2 width" },
  { 0x1e, "Energy window #3 center" },
  { 0x1f, "Energy window #3 width" },
  { 0x20, "Energy window #4 center" },
  { 0x21, "Energy window #4 width" },
  { 0x22, "??Energy window #5 center" },
  { 0x23, "??Energy window #5 width" },
  { 0x24, "Patient orientation" },
  { 0x25, "Spatial resolution" },
  { 0x26, "Slice thickness" },
  { 0x27, "Image X dimension" },
  { 0x28, "Image Y dimension" },
  { 0x29, "Image Z dimension" },
  { 0x2a, "Image pixel width" },
  { 0x2b, "Uniformity corr. file" },
  { 0x2c, "Acquisition zoom factor" },
  { 0x2d, "Total counts in set" },
  { 0x2e, "Time / frame" },
  { 0x2f, "Total acq. time" },
  { 0x30, "Maximum pixel value" },
  { 0x31, "Minimum pixel value" },
  { 0x32, "R-R interval time" },
  { 0x33, "Percent of cycle imaged" },
  { 0x34, "# of cycles accepted" },
  { 0x35, "# of cycles rejected" },
  { 0x36, "Approximate ED frame" },
  { 0x37, "Approximate ES frame" },
  { 0x38, "Approximate EF" },
  { 0x39, "Starting angle" },
  { 0x3a, "Degrees of rotation" },
  { 0x3b, "Direction of rotation" },
  { 0x3c, "Cont. or step/shoot" },
  { 0x3d, "Lim recon start frame" },
  { 0x3e, "Upper window grey shade" },
  { 0x3f, "Lower lvl grey shade" },
  { 0x40, "Associated color map" },
  { 0x41, "Custom color map file" },
  { 0x42, "Manipulated image" },
  { 0x43, "Axis of rotation corr." },
  { 0x44, "Reorientation azimuth" },
  { 0x45, "Reorientation elevation" },
  { 0x46, "Filter type" },
```

```
  { 0x47, "Filter order" },
  { 0x48, "Filter cutoff frequency" },
  { 0x49, "Reconstruction type" },
  { 0x4a, "Attenuation coefficient" },
  { 0x4b, "Associated parent file" },
  { 0x4c, "Unique patient key" },
  { 0x52, "Normalization crv file" },
  { 0x53, "Unique object key" },
  { 0x54, "This phase of VFR is" },
  { 0x55, "True color value" },
  { 0x56, "# of sets of x,y,z grps" },
  { 0x57, "Scale factor of set" },
  { 0x6d, "Date of birth" },
  { 0x6e, "Directional orientation" },
  { 0x6f, "Number of VFR studies" },
  { 0x70, "R-R low tolerance" },
  { 0x71, "R-R high tolerance" },
  { 0x72, "Prog specific results:" },

  { 0x99, NULL }
};

void printname( int , int , uint16_t v )
{
  if( v == 0x1 )
    {
    std::cout << "DATABASE PARAMETERS" << std::endl;
    std::cout << "____________________" << std::endl;
    }
  else if( v == 0x27 )
    {
    std::cout << "IMAGE PARAMETERS" << std::endl;
    std::cout << "_________________" << std::endl;
    }
  else if( v == 0x13 )
    {
    std::cout << "EXTRA PARAMETERS" << std::endl;
    std::cout << "_________________" << std::endl;
    }
  else if( v == 0x2e )
    {
    std::cout << "*** NOT CURRENTLY USED :" << std::endl;
    }
  static const unsigned int n = sizeof( Array ) / sizeof( *Array ) - 1;
  for( unsigned int i = 0; i < n; ++i )
    {
    if( v == Array[i].key )
      {
      std::cout << /*"" << std::dec << len << "," << mult << " " << */ Array[i]
      .name;
      std::cout << " : ";
      return;
      }
    }
  std::cout << /*"\t# " << std::dec << len << "," << mult << */ std::hex << v <
      < "\t: ";
}

uint16_t readint16(std::istream &is )
{
  uint16_t val;
  is.read( (char*)&val, sizeof( val ));
  return (val>>8) | (val<<8);
}

uint32_t readint32(std::istream &is )
{
  uint32_t val;
  is.read( (char*)&val, sizeof( val ));
  val= ((val<<8)&0xFF00FF00) | ((val>>8)&0x00FF00FF);
  return (val>>16) | (val<<16);
}

float readfloat32(std::istream &is )
{
  union { uint32_t val; float f;} dual;
  dual.val = readint32(is);
  return dual.f;
}

struct el
```

```
{
  uint16_t v1;
  uint16_t v2;
  uint16_t v3;
  void read( std::istream & is )
    {
    v1 = readint16(is);
    v2 = readint16(is);
    v3 = readint16(is);
    }
  void print( std::ostream & os )
    {
    os << std::hex << v1 << "\t" << v2 << "\t" << v3 << std::endl;
    }
};

std::vector<el> Vel;

void readelement( std::istream & is )
{
  el e;
  e.read( is );
  Vel.push_back( e );
}

void printascii( uint16_t tag, const char *buffer, size_t len )
{
  std::ostream & os = std::cout;
  if( tag == 0x72 )
    {
    os << "\n  ";
    for(size_t i = 0; i < len; ++i)
      {
      const char &c = buffer[i];
      if( c == 0x0 ) os << "!";
      else if( c == 0x0f ) os << " ";
      else if( c == 0x17 ) os << ":";
      else if( c == 0x14 ) os << ":";
      else if( c == 0x10 ) os << ":";
      else if( c == 0x16 ) os << ":";
      else if( c == 0x08 ) os << ":";
      else if( c == 0x0b ) os << ":";
      else if( c == 0x0e ) os << ":";
      else if( c == 0x07 ) os << ":";
      else os << c;
      }
    os << "";
    }
  else
    {
    (void)len;
    os << "" << buffer << "";
    }
}

bool DumpADAC( std::istream & is )
{
  std::ostream &os = std::cout;

  char magic[6 + 1];
  magic[6] = 0;
  is.read( magic, 6);
//  std::cout << magic << " ";
  assert( strcmp( magic, "adac01" ) == 0 );
  char c = is.get();
  assert( c == 0 );
  c = is.get();
  assert( c == 'X' );

  uint16_t v;
  v = readint16(is);
//  std::cout << v << std::endl;
  assert( v == 512 ); // ??

  int nel = 87;
  for (int i = 0; i <= nel; ++i )
    {
    readelement( is );
    }

  char buffer[512];
```

```
  for( int i = 0; i <= nel; ++i )
    {
    const el &e = Vel[i];
    int diff;
    if( i == nel )
      {
      diff = 2048 - e.v3;
      if( diff > 512 ) diff = 512;
      }
    else
      {
      const el &enext = Vel[i+1];
      diff = enext.v3 - e.v3;
      }
    is.seekg( e.v3, std::ios::beg );
    //std::cout << "(" << std::hex << std::setw( 2 ) << std::setfill( '0' ) <<
       e.v1 << ") " << std::hex << std::setw( 3 ) << std::setfill( '0' ) << e.v2 << "
       ";
    printname( diff, 0, e.v1 );
    int mult = 1;
    if( e.v2 == 0 )
      {
      is.read( buffer, diff);
      buffer[ diff ] = 0;
      printascii( e.v1, buffer, diff);
      }
    else if( e.v2 == 0x100 )
      {
      mult = diff / 2;
      assert( diff == 2 * mult );
      for ( int ii = 0; ii < mult; ++ii )
        {
        if ( ii ) os << "\\";
        uint16_t val = readint16(is);
        os << "" << std::dec << val << "";
        }
      }
    else if( e.v2 == 0x200 )
      {
      assert( diff == 4 );
      uint32_t val = readint32(is);
      os << "" << std::dec << val << "";
      }
    else if( e.v2 == 0x300 )
      {
      assert( diff == 4 );
      float val = readfloat32(is);
      os << "" << std::dec << val << "";
      }
     else
      {
      assert( 0 );
      }
    os << std::endl;
    }
  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet
      ();

  // (0019,1061) UN (OB) 61\64\61\63\30         # 2048,1 Ver200 ADAC Pegasys
       Headers
  const gdcm::PrivateTag tver200adacpegasysheaders(0x0019,0x61,
      "ADAC_IMG");
  if( !ds.FindDataElement( tver200adacpegasysheaders ) ) return
      1;
  const gdcm::DataElement& ver200adacpegasysheaders = ds.
      GetDataElement( tver200adacpegasysheaders );
  if ( ver200adacpegasysheaders.IsEmpty() ) return 1;
```

```
  const gdcm::ByteValue * bv = ver200adacpegasysheaders.
      GetByteValue();

  // (0019,1021) US 1               # 2,1 Ver200 Number of ADAC Headers
  // TODO

  // (0019,1041) IS [2048\221184 ]  # 12,1-n Ver200 ADAC Header/Image Size
  if( bv->GetLength() != 2048 ) return 1;

  gdcm::Element<gdcm::VR::IS,gdcm::VM::VM2>
       el;
  const gdcm::PrivateTag tver200adacheaderimagesize(0x0019,0x41
      ,"ADAC_IMG");
  if( !ds.FindDataElement( tver200adacheaderimagesize ) ) return
       1;
  const gdcm::DataElement& ver200adacheaderimagesize = ds.
      GetDataElement( tver200adacheaderimagesize );
  el.SetFromDataElement( ver200adacheaderimagesize );
  if( el.GetValue(0) != 2048 ) return 1;

  std::istringstream is;
  std::string dup( bv->GetPointer(), bv->GetLength() );
  is.str( dup );
  bool b = DumpADAC( is );
  if( !b ) return 1;



  return 0;
}
```

27.33 DumpGEMSMovieGroup.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

bool PrintNameValueMapping( gdcm::SequenceOfItems *
      sqi_values,
gdcm::SequenceOfItems *sqi_names, std::string const &
      indent )
{
  using namespace gdcm;
  // prepare names mapping:
  typedef VRToType<VR::UL>::Type UL;
  std::map< UL, std::string > names;
  assert( sqi_names );
  assert( sqi_values );
  SequenceOfItems::SizeType s = sqi_names->
      GetNumberOfItems();
  PrivateTag tindex(0x7fe1,0x71,"GEMS_Ultrasound_MovieGroup_001");
  PrivateTag tname (0x7fe1,0x72,"GEMS_Ultrasound_MovieGroup_001");
  // First sequence contains all possible names (this is a dict)
  for( SequenceOfItems::SizeType i = 1; i <= s; ++i )
    {
    const Item & item = sqi_names->GetItem( i );
```

```
    const DataSet & ds = item.GetNestedDataSet();
    if( !ds.FindDataElement( tindex )
      || !ds.FindDataElement( tname ) )
      {
      return false;
      }
    const DataElement & index = ds.GetDataElement(
      tindex );
    const DataElement & name = ds.GetDataElement(
      tname );
    if( index.IsEmpty() || name.IsEmpty() )
      {
      return false;
      }
    gdcm::Element<VR::UL, VM::VM1> el1;
    el1.SetFromDataElement( index );

    gdcm::Element<VR::LO, VM::VM1> el2;
    el2.SetFromDataElement( name );
//    std::cout << el1.GetValue() << " " << el2.GetValue() << std::endl;
    names.insert( std::make_pair< UL, std::string > ( el1.GetValue(),
      el2.GetValue() ) );
    }

  SequenceOfItems::SizeType s2 = sqi_values->
      GetNumberOfItems();
  assert( s2 <= s );
  PrivateTag tindex2(0x7fe1,0x48,"GEMS_Ultrasound_MovieGroup_001");
  for( SequenceOfItems::SizeType i = 1; i <= s2; ++i )
    {
    const Item & item = sqi_values->GetItem( i );
    const DataSet & ds = item.GetNestedDataSet();
    if( !ds.FindDataElement( tindex2 ) )
      {
      return false;
      }
    const DataElement & index2 = ds.GetDataElement(
      tindex2 );
    if( index2.IsEmpty() )
      {
      return false;
      }
    gdcm::Element<VR::FD, VM::VM1_2> el1;
    el1.SetFromDataElement( index2 );

    UL copy = el1.GetValue();
#if 1
    std::cout << indent;
    std::cout << "( " << names[ copy ];
#endif
    // (7fe1,1052) FD 1560                                      # 8,1 ?
    // (7fe1,1057) LT [MscSkelSup]                              # 10,1 ?
    //PrivateTag tvalue(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001");
    PrivateTag tvalueint(0x7fe1,0x49,"GEMS_Ultrasound_MovieGroup_001"
      ); // UL
    PrivateTag tvaluefloat1(0x7fe1,0x51,"
      GEMS_Ultrasound_MovieGroup_001"); // FL
    PrivateTag tvaluefloat(0x7fe1,0x52,"
      GEMS_Ultrasound_MovieGroup_001"); // FD
    PrivateTag tvalueul(0x7fe1,0x53,"GEMS_Ultrasound_MovieGroup_001")
      ; // UL
    PrivateTag tvaluesl(0x7fe1,0x54,"GEMS_Ultrasound_MovieGroup_001")
      ; // SL
    PrivateTag tvalueob(0x7fe1,0x55,"GEMS_Ultrasound_MovieGroup_001")
      ; // OB
    PrivateTag tvaluetext(0x7fe1,0x57,"GEMS_Ultrasound_MovieGroup_001
      ");  // LT
    PrivateTag tvaluefd(0x7fe1,0x77,"GEMS_Ultrasound_MovieGroup_001")
      ;  // FD / 1-N
    PrivateTag tvaluesl3(0x7fe1,0x79,"GEMS_Ultrasound_MovieGroup_001"
      );  // SL / 1-N
    PrivateTag tvaluesl2(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001"
      );  // SL ??
    PrivateTag tvaluefd1(0x7fe1,0x87,"GEMS_Ultrasound_MovieGroup_001"
      );  // FD / 1-N
    PrivateTag tvaluefloat2(0x7fe1,0x88,"
      GEMS_Ultrasound_MovieGroup_001");  // FD ??
#if 1
    std::cout << " ) = ";
#endif
    if( ds.FindDataElement( tvalueint ) )
```

```
      {
      const DataElement & value = ds.GetDataElement(
      tvalueint );
      gdcm::Element<VR::UL,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout << el2.GetValue() << std::endl;
      }
    else if( ds.FindDataElement( tvaluefloat1 ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvaluefloat1 );
      gdcm::Element<VR::FL,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout << el2.GetValue() << std::endl;
      }
    else if( ds.FindDataElement( tvaluefloat ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvaluefloat );
      gdcm::Element<VR::FD,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout << el2.GetValue() << std::endl;
      }
    else if( ds.FindDataElement( tvaluesl ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvaluesl );
      gdcm::Element<VR::SL,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout << el2.GetValue() << std::endl;
      }
    else if( ds.FindDataElement( tvalueul ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvalueul );
      gdcm::Element<VR::UL,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      assert( el2.GetLength() == 1 );
      std::cout << el2.GetValue() << std::endl;
      }
    else if( ds.FindDataElement( tvalueob ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvalueob );
//      gdcm::Element<VR::SL,VM::VM1> el2;
//      el2.SetFromDataElement( value );
//      std::cout << el2.GetValue() << std::endl;
      std::cout << value << std::endl;
      }
    else if( ds.FindDataElement( tvaluetext ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvaluetext );
      gdcm::Element<VR::LT,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout << el2.GetValue() << std::endl;
      }
    else if( ds.FindDataElement( tvaluesl2 ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvaluesl2 );
      gdcm::Element<VR::SL,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
      assert( el2.GetLength() == 4 );
      std::cout << std::endl;
      }
    else if( ds.FindDataElement( tvaluesl3 ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvaluesl3 );
      gdcm::Element<VR::SL,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
//      assert( el2.GetLength() == 4 );
      std::cout << std::endl;
      }
    else if( ds.FindDataElement( tvaluefd ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvaluefd );
```

```
      gdcm::Element<VR::FD,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
//      assert( el2.GetLength() == 4 || el2.GetLength() == 3 || el2.GetLength()
       == 8 );
      std::cout << std::endl;
      }
    else if( ds.FindDataElement( tvaluefloat2 ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvaluefloat2 );
      gdcm::Element<VR::FD,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
      assert( el2.GetLength() == 2 );
      std::cout << std::endl;
      }
    else if( ds.FindDataElement( tvaluefd1 ) )
      {
      const DataElement & value = ds.GetDataElement(
      tvaluefd1 );
      gdcm::Element<VR::FD,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
      assert( el2.GetLength() == 4 );
      std::cout << std::endl;
      }
    else
      {
      std::cout << "(no value)" << std::endl;
//      std::cout << ds << std::endl;
      assert( ds.Size() == 2 );
      }
    }
  return true;
}

bool PrintNameValueMapping2( gdcm::PrivateTag const & privtag,
      const gdcm::DataSet & ds ,
gdcm::SequenceOfItems *sqi_names, std::string const &
      indent )
{
    if( !ds.FindDataElement( privtag ) ) return 1;
    const gdcm::DataElement& seq_values = ds.GetDataElement
      ( privtag );
    gdcm::SmartPointer<gdcm::SequenceOfItems>
       sqi = seq_values.GetValueAsSQ();

  return PrintNameValueMapping( sqi, sqi_names, indent);
}

bool PrintNameValueMapping3( gdcm::PrivateTag const & privtag1,
       gdcm::PrivateTag const & privtag2, const gdcm::DataSet
       & ds ,
gdcm::SequenceOfItems *sqi_names, std::string const &
      indent )
{
  if( !ds.FindDataElement( privtag1 ) )
    {
    assert( 0 );
    return false;
    }
  const gdcm::DataElement& values10name = ds.GetDataElement
      ( privtag1 );
  gdcm::Element<gdcm::VR::LO,gdcm::VM::VM1>
       el;
  el.SetFromDataElement( values10name );
  std::cout << std::endl;
  std::cout << " <" << el.GetValue().c_str() << ">" << std::endl;

  return PrintNameValueMapping2( privtag2, ds, sqi_names, indent);
}

bool print73( gdcm::DataSet const & ds10, gdcm::SequenceOfItems
       *sqi_dict, std::string const & indent )
{
  const gdcm::PrivateTag tseq_values73(0x7fe1,0x73,"
      GEMS_Ultrasound_MovieGroup_001");
  if( !ds10.FindDataElement( tseq_values73 ) )
    {
    std::cout << indent << "No group 73" << std::endl;
```

```
    return false;
    }
  const gdcm::DataElement& seq_values73 = ds10.GetDataElement
      ( tseq_values73 );
  gdcm::SmartPointer<gdcm::SequenceOfItems>
       sqi_values73 = seq_values73.GetValueAsSQ();

  int ni3 = sqi_values73->GetNumberOfItems();
  for( int i3 = 1; i3 <= ni3; ++i3 )
    {
    gdcm::Item &item_73 = sqi_values73->GetItem(i3);
    gdcm::DataSet &ds73 = item_73.GetNestedDataSet
      ();
    assert( ds73.Size() == 3 );

    const gdcm::PrivateTag tseq_values74name(0x7fe1,0x74,"
      GEMS_Ultrasound_MovieGroup_001");
    const gdcm::PrivateTag tseq_values75(0x7fe1,0x75,"
      GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values74name, tseq_values75, ds73, sqi_dict,
      indent);
    std::cout << std::endl;
    }
  return true;
}

bool print83( gdcm::DataSet const & ds10, gdcm::SequenceOfItems
       *sqi_dict, std::string const & indent )
{
  const gdcm::PrivateTag tseq_values83(0x7fe1,0x83,"
      GEMS_Ultrasound_MovieGroup_001");
  if( !ds10.FindDataElement( tseq_values83 ) )
    {
    std::cout << indent << "No group 83" << std::endl;
    return false;
    }
  const gdcm::DataElement& seq_values83 = ds10.GetDataElement
      ( tseq_values83 );
  gdcm::SmartPointer<gdcm::SequenceOfItems>
       sqi_values83 = seq_values83.GetValueAsSQ();

  int ni3 = sqi_values83->GetNumberOfItems();
  for( int i3 = 1; i3 <= ni3; ++i3 )
    {
    gdcm::Item &item_83 = sqi_values83->GetItem(i3);
    gdcm::DataSet &ds83 = item_83.GetNestedDataSet
      ();
    assert( ds83.Size() == 3 );

    const gdcm::PrivateTag tseq_values84name(0x7fe1,0x84,"
      GEMS_Ultrasound_MovieGroup_001");
    const gdcm::PrivateTag tseq_values85(0x7fe1,0x85,"
      GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values84name, tseq_values85, ds83, sqi_dict,
      indent);
    std::cout << std::endl;
    }
  return true;
}

bool PrintNameValueMapping4( gdcm::PrivateTag const & privtag0,
       const gdcm::DataSet & subds, gdcm::PrivateTag
      const & privtag1, gdcm::PrivateTag const & privtag2,
gdcm::SequenceOfItems *sqi_dict, std::string const &
      indent )
{
  if( !subds.FindDataElement( privtag0 ) )
    {
    assert( 0 );
    return 1;
    }
  const gdcm::DataElement& seq_values10 = subds.GetDataElement
      ( privtag0 );
  gdcm::SmartPointer<gdcm::SequenceOfItems>
       sqi_values10 = seq_values10.GetValueAsSQ();

  int ni1 = sqi_values10->GetNumberOfItems();
//  assert( ni1 == 1 );
  for( int i1 = 1; i1 <= ni1; ++i1 )
    {
    gdcm::Item &item_10 = sqi_values10->GetItem(i1);
```

```
    gdcm::DataSet &ds10 = item_10.GetNestedDataSet
      ();
    assert( ds10.Size() == 2 + 3 );
    // (7fe1,0010)
    // (7fe1,1012)
    // (7fe1,1018)
    // (7fe1,1020)
    // (7fe1,1083)

    PrintNameValueMapping3( privtag1, privtag2, ds10, sqi_dict, "  " );
    std::cout << std::endl;

    const gdcm::PrivateTag tseq_values20(0x7fe1,0x20,"
      GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values20 ) )
      {
      assert( 0 );
      return 1;
      }
    const gdcm::DataElement& seq_values20 = ds10.
      GetDataElement( tseq_values20 );
    gdcm::SmartPointer<gdcm::SequenceOfItems>
       sqi_values20 = seq_values20.GetValueAsSQ();

    int ni2 = sqi_values20->GetNumberOfItems();
    //assert( ni == 1 );
    for( int i2 = 1; i2 <= ni2; ++i2 )
      {
      gdcm::Item &item_20 = sqi_values20->GetItem(i2);
      gdcm::DataSet &ds20 = item_20.GetNestedDataSet
      ();
      int count = ds20.Size();
      assert( ds20.Size() == 2 + 3 || ds20.Size() == 2 + 2 );
      // (7fe1,0010)
      // (7fe1,1024)
      // (7fe1,1026)
      // (7fe1,1036)
      // (7fe1,1083) (*)

      const gdcm::PrivateTag tseq_values20name(0x7fe1,0x24,"
      GEMS_Ultrasound_MovieGroup_001");
      const gdcm::PrivateTag tseq_values26(0x7fe1,0x26,"
      GEMS_Ultrasound_MovieGroup_001");
      PrintNameValueMapping3( tseq_values20name, tseq_values26, ds20, sqi_dict,
       "   ");
      std::cout << std::endl;

      print83(ds20, sqi_dict, "    ");
      }

    print83(ds10, sqi_dict, "   ");
    }
  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  reader.Read();

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");

  if( !ds.FindDataElement( tseq ) ) return 1;
  const DataElement& seq = ds.GetDataElement( tseq );

  SmartPointer<SequenceOfItems> sqi = seq.
      GetValueAsSQ();
  assert( sqi->GetNumberOfItems() == 1 );

  Item &item = sqi->GetItem(1);
  DataSet &subds = item.GetNestedDataSet();

  const PrivateTag tseq_dict(0x7fe1,0x70,"
      GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq_dict ) ) return 1;
```

```
  const DataElement& seq_dict = subds.GetDataElement(
      tseq_dict );
  SmartPointer<SequenceOfItems> sqi_dict =
      seq_dict.GetValueAsSQ();

  const PrivateTag tseq_values8(0x7fe1,0x8,"
      GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq_values8 ) ) return 1;
  const DataElement& seq_values8 = subds.GetDataElement
      ( tseq_values8 );
  SmartPointer<SequenceOfItems> sqi_values8 =
      seq_values8.GetValueAsSQ();

  const PrivateTag tseq_values8name(0x7fe1,0x2,"
      GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq_values8name ) ) return 1;
  const DataElement& values8name = subds.GetDataElement
      ( tseq_values8name );
{
  Element<VR::LO,VM::VM1> el;
  el.SetFromDataElement( values8name );
  std::cout << el.GetValue() << std::endl;
}
  int count = subds.Size();
  assert( subds.Size() == 3 + 2 + 1 || subds.Size() == 3 + 2 + 2);

//  (7fe1,0010) # 30,1 Private Creator
//  (7fe1,1002) # 8,1 US MovieGroup Value 0008 Name
//  (7fe1,1003) # 4,1 ?
//  (7fe1,1008) # 8140,1 US MovieGroup Value 0008 Sequence
//  (7fe1,1010) # 1372196,1 ?
//  (7fe1,1070) # 33684,1 US MovieGroup Dict
//  (7fe1,1073) (*)
  PrintNameValueMapping( sqi_values8, sqi_dict, " ");

  const PrivateTag tseq_values10(0x7fe1,0x10,"
      GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq_values10name(0x7fe1,0x12,"
      GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq_values18(0x7fe1,0x18,"
      GEMS_Ultrasound_MovieGroup_001");
  PrintNameValueMapping4( tseq_values10, subds, tseq_values10name,
      tseq_values18, sqi_dict,"  ");

  print73( subds, sqi_dict, "  " );

#if 0
  gdcm::DataSet::ConstIterator it = subds.Begin
      ();
  for( ; it != subds.End(); ++it )
    {
  const gdcm::DataElement &de = *it;
    std::cout << de.GetTag() << std::endl;
    }
#endif

  return 0;
}
```

27.34 DumpToSQLITE3.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Ref:
 * http://massmail.spl.harvard.edu/public-archives/slicer-devel/2010/
```

```
      004408.html
 *
 * Implementation details:
 * http://www.sqlite.org/c3ref/bind_blob.html
 * http://www.adp-gmbh.ch/sqlite/bind_insert.html
 */
#include "gdcmScanner.h"
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include "gdcmTrace.h"

#include "sqlite3.h"

#include <stdio.h>
#include <time.h>

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  time_t time_start = time(0);

  gdcm::Trace::SetDebug( false );
  gdcm::Trace::SetWarning( false );
  const char *inputdirectory = argv[1];

  gdcm::Directory d;
  unsigned int nfiles = d.Load( inputdirectory, true);

  gdcm::Scanner s;
  using gdcm::Tag;
  s.AddTag( Tag(0x20,0xd) ); // Study Instance UID
  s.AddTag( Tag(0x20,0xe) ); // Series Instance UID

  bool b = s.Scan( d.GetFilenames() );
  if( !b ) return 1;
  time_t time_scanner = time(0);

  std::cout << "Finished loading data from : " << nfiles << " files" <<
      std::endl;

//  MappingType const &mappings = s.GetMappings();


  sqlite3* db;
  sqlite3_open("./dicom.db", &db);

  if(db == 0)
    {
    std::cerr << "Could not open database." << std::endl;
    return 1;
    }

  const char sql_stmt[] = "create table browser (seriesuid, studyuid)";
  int   ret;

  char *errmsg;
  ret = sqlite3_exec(db, sql_stmt, 0, 0, &errmsg);

  if(ret != SQLITE_OK)
    {
    printf("Error in statement: %s [%s].\n", sql_stmt, errmsg);
    return 1;
    }
  using gdcm::Directory;
  using gdcm::Scanner;
  const Directory::FilenamesType& files = d.GetFilenames();
  Directory::FilenamesType::const_iterator file = files.begin();

  sqlite3_stmt *stmt;
  if ( sqlite3_prepare(
      db,
      "insert into browser values (?,?)",  // stmt
      -1, // If than zero, then stmt is read up to the first nul terminator
      &stmt,
      0  // Pointer to unused portion of stmt
  )
    != SQLITE_OK)
    {
```

```
    printf("\nCould not prepare statement.");
    return 1;
    }
  //printf("\nThe statement has %d wildcards\n",
      sqlite3_bind_parameter_count(stmt));
  for(; file != files.end(); ++file)
    {
    const char *filename = file->c_str();
    bool b = s.IsKey(filename);
    if( b )
      {
      const Scanner::TagToValue &mapping = s.GetMapping(filename);
      Scanner::TagToValue::const_iterator it = mapping.begin();

      sqlite3_reset(stmt);

      for( int index = 1; it != mapping.end(); ++it, ++index)
        {
        //const Tag & tag = it->first;
        const char *value = it->second;

        if (sqlite3_bind_text (
            stmt,
            index,  // Index of wildcard
            value,
            strlen(value),  // length of text
            SQLITE_STATIC // SQLite assumes that the information is in static
        )
          != SQLITE_OK)
          {
          printf("\nCould not bind int.\n");
          return 1;
          }
        }
      if (sqlite3_step(stmt) != SQLITE_DONE)
        {
        printf("\nCould not step (execute) stmt.\n");
        return 1;
        }
      }
    }

  sqlite3_close(db);

  time_t time_sqlite = time(0);

  std::cout << "Time to scan DICOM files: " << (time_scanner - time_start) <<
      std::endl;
  std::cout << "Time to build SQLITE3: " << (time_sqlite - time_scanner) <<
      std::endl;

  return 0;
}
```

27.35 DuplicatePCDE.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
/*
```

```
  Usage:
  DuplicatePCDE gdcmData/D_CLUNIE_CT1_J2KI.dcm out.dcm

aka:
medical.nema.org/medical/dicom/DataSets/WG04/IMAGES/J2KI/CT1_J2KI

See:
  gdcmConformanceTests/CT1_J2KI_DuplicatePCDE.dcm

Original thread can be found at:

http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/
      82f28c4db28963af


Question:
1.
  There is no restriction for a specific Private Creator Data Element
(PCDE) to be unique within the same group, right ?
  Decoders of Private Data would have to handle the case where a PCDE
would be repeated and should NOT stop on the first instance of a
particular PCDE, right ?

  Eg. when searching for the tag associated with
(0x0029,0x0010,"SIEMENS CSA HEADER") in the following (pseudo)
dataset:

(0029,0010) LO [SIEMENS CSA HEADER]                     #  18, 1
PrivateCreator
(0029,0011) LO [SIEMENS MEDCOM HEADER]                  #  22, 1
PrivateCreator
(0029,0012) LO [SIEMENS MEDCOM HEADER2]                 #  22, 1
PrivateCreator
(0029,0013) LO [SIEMENS CSA HEADER]                     #  18, 1
PrivateCreator
(0029,1008) CS [IMAGE NUM 4]                            #  12, 1
CSAImageHeaderType
(0029,1009) LO [20050723]                               #   8, 1
CSAImageHeaderVersion
(0029,1010) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo
(0029,1018) CS [MR]                                     #   2, 1
CSASeriesHeaderType
(0029,1019) LO [20050723]                               #   8, 1
CSASeriesHeaderVersion
(0029,1020) OB 53\56\31\30\04\03\02\01\2c\00\00\00\4d
\00\00\00\55\73\65\64\50\61... # 51520, 1 CSASeriesHeaderInfo
(0029,1131) LO [4.0.163088300]                          #  14, 1
PMTFInformation1
(0029,1132) UL 32768                                    #   4, 1
PMTFInformation2
(0029,1133) UL 0                                        #   4, 1
PMTFInformation3
(0029,1134) CS [DB TO DICOM]                            #  12, 1
PMTFInformation4
(0029,1260) ?? 63\6f\6d\20                              #   4, 1
Unknown Tag & Data
(0029,1310) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo

  one should return two instances, correct ?

Answer:
I would say that this is covered in principle by the PS 3.5 7.1
"The Data Elements ... shall occur at most once in a Data Set"
rule, since the data element is defined by the tuple
(private creator,gggg,ee) where xxee is the element
number and xx is arbitrary and has no inherent meaning and
does not serve to disambiguate the data element.

E.g.:

(0019,0030) Private Creator ID = "Smith"
...
(0019,0032) Private Creator ID = "Smith"
...
(0019,3015) Fractal Index = "32"
...
(0019,3215) Fractal Index = "32"

would be illegal because even though they are assigned different
```

```
(completely arbitrary) blocks, with the same group, element
number and private creator, (0019,3015) and (0019,3215) are the
"same" data element.

*/

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  // Let's get all private element from group 0x9:
/*
(0009,0010) LO [GEMS_IDEN_01]                                      # 12,1
       Private Creator
(0009,1001) LO [GE_GENESIS_FF ]                                    # 14,1 Full
       fidelity
(0009,1002) SH [CT01]                                              # 4,1 Suite
       id
(0009,1004) SH [HiSpeed CT/i]                                      # 12,1
       Product id
(0009,1027) SL 862399669                                           # 4,1 Image
       actual date
(0009,1030) SH (no value)                                          # 0,1 Service
       id
(0009,1031) SH (no value)                                          # 0,1 Mobile
       location number
(0009,10e6) SH [05]                                                # 2,1 Genesis
       Version - now
(0009,10e7) UL 973283917                                           # 4,1 Exam
       Record checksum
(0009,10e9) SL 862399669                                           # 4,1 Actual
       series data time stamp
*/
  gdcm::Tag start(0x0009,0x0);
  // Create a temporary duplicate dataset, since we cannot insert data element
       as we go over them (std::set
  // would reorganize itself as we go over it ...)
  gdcm::DataSet dup;
  gdcm::Tag new_private(0x0009,0x0);
  while (start.GetGroup() == 0x9 )
    {
    const gdcm::DataElement& de = ds.FindNextDataElement
      (start);
    const gdcm::Tag &t = de.GetTag();
    if( t.IsPrivateCreator() )
      {
      std::cout << t << std::endl;
      // Ok let's duplicate into the next available attribute:
      gdcm::DataElement duplicate = de;
      duplicate.GetTag().SetElement( t.GetElement() +
       1 );
      dup.Insert( duplicate );
      new_private = duplicate.GetTag();
      }
    else if( t.IsPrivate() && !t.IsPrivateCreator() )
      {
      //std::cout << de << std::endl;
      std::string owner = ds.GetPrivateCreator( de.GetTag
      () );
      //std::cout << owner << std::endl;
      gdcm::DataElement duplicate = de;
      duplicate.GetTag().SetPrivateCreator( new_private
      );
      if( const gdcm::ByteValue *bv = duplicate.GetByteValue
      () )
        {
```

```
        // Warning: when doing : duplicate = de, only the pointer to the
       ByteValue is passed
        // (to avoid large memory duplicate). We need to explicitely duplicate
       the bytevalue ourselves:
        gdcm::ByteValue *dupbv = new gdcm::ByteValue
      ( bv->GetPointer(),
          bv->GetLength() );
        // Let's recognize the duplicated ASCII-type elements:
        if( duplicate.GetVR() & gdcm::VR::VRASCII )
          dupbv->Fill( 'X' );
        duplicate.SetValue( *dupbv );
        }
      dup.Insert( duplicate );
      }
    start = t;
    // move to next possible 'public' element
    start.SetElement( start.GetElement() + 1);
    }

  gdcm::DataSet::ConstIterator it = dup.Begin(
      );
  for( ; it != dup.End(); ++it )
    {
    ds.Insert( *it );
    }

  gdcm::Writer w;
  w.SetFile( file );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.36 ELSCINT1WaveToText.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"

/*
 * This example shows how to read a Wave Information tag from ELSCINT1
 * The wave information is stored in Tag (01e1,18,ELSCINT1) hidden in a
 * Secondary Capture Image Storage (usually a 'N' Symbol is shown)
 *
 * Everything done in this code is for the sole purpose of writing
       interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Gauthier Bouilhol
 */

template <typename T>
bool dumpargs(std::ostream & os, T c1, T c2, T c3, T c4, T c5, T c6, T c7, T c8
      )
```

```
{
  static const char sep = '\t';
  os << c1 << sep << c2 << sep << c3 << sep << c4 << sep << c5 << sep << c6 <<
      sep << c7 << sep << c8;
  os << std::endl;
  return true;
}

bool wave2stream( std::ostream &text_file, const char *in, size_t len )
{
  short * buffer = (short*)in;
  int length = len / sizeof( short );
  text_file << "COMPLETE_WAVE" << '\t' << "MASK"       << '\t' << "
      AQUISITION_PROFIL" << '\t' << "END-INHALE" << '\t' << "END-EXHALE" << '\t' << "
      AQUISITION_WAVE" << '\t' << "WAVE_STATISTICS" << '\t' << "MASK"        << std::endl;
  for (int i=0;i<length-76;i+=2)
    {
    if ( i < 74 )
      {
      if (buffer[i+75] == 0)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 0
                  << '\t' << "  "         << '\t' << "  "         << '\t' << "  "
                << '\t' << buffer[i]         << '\t' << buffer[i+1] << std::endl;
      if (buffer[i+75] == 16384)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 0
                  << '\t' << buffer[i+74] << '\t' << "  "         << '\t' << "  "
                << '\t' << buffer[i]         << '\t' << buffer[i+1] << std::endl;
      if (buffer[i+75] == 256)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 0
                  << '\t' << "  "         << '\t' << buffer[i+74] << '\t' << "  "
                << '\t' << buffer[i]         << '\t' << buffer[i+1] << std::endl;
      if (buffer[i+75] == -32768)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 1
                  << '\t' << "  "         << '\t' << "  "         << '\t' << buffer[i
      +74]      << '\t' << buffer[i]         << '\t' << buffer[i+1] << std::endl;
      if (buffer[i+75] == -16384)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 1
                  << '\t' << buffer[i+74] << '\t' << "  "         << '\t' << buffer[i
      +74]      << '\t' << buffer[i]         << '\t' << buffer[i+1] << std::endl;
      if (buffer[i+75] == -32512)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 1
                  << '\t' << "  "         << '\t' << buffer[i+74] << '\t' << buffer[i
      +74]      << '\t' << buffer[i]         << '\t' << buffer[i+1] << std::endl;
      }
    else
      {
      if (buffer[i+75] == 0)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 0
                  << '\t' << "  "         << '\t' << "  "         << '\t' << "  "
                << '\t' << "  "              << '\t' << "  "        << std::endl;
      if (buffer[i+75] == 16384)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 0
                  << '\t' << buffer[i+74] << '\t' << "  "         << '\t' << "  "
                << '\t' << "  "              << '\t' << "  "        << std::endl;
      if (buffer[i+75] == 256)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 0
                  << '\t' << "  "         << '\t' << buffer[i+74] << '\t' << "  "
                << '\t' << "  "              << '\t' << "  "        << std::endl;
      if (buffer[i+75] == -32768)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 1
                  << '\t' << "  "         << '\t' << "  "         << '\t' << buffer[i
      +74]      << '\t' << "  "              << '\t' << "  "        << std::endl;
      if (buffer[i+75] == -16384)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 1
                  << '\t' << buffer[i+74] << '\t' << "  "         << '\t' << buffer[i
      +74]      << '\t' << "  "              << '\t' << "  "        << std::endl;
      if (buffer[i+75] == -32512)
        text_file << buffer[i+74]    << '\t' << buffer[i+75] << '\t' << 1
                  << '\t' << "  "         << '\t' << buffer[i+74] << '\t' << buffer[i
      +74]      << '\t' << "  "              << '\t' << "  "        << std::endl;
      }
    }

  return true;
}

int main(int argc, char *argv [])
{
  if( argc < 3 ) return 1;
  const char *filename = argv[1];
  const char *outfilename = argv[2];
```

```
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet
      ();

  const gdcm::PrivateTag twave(0x01e1,0x18,"ELSCINT1");
  if( !ds.FindDataElement( twave ) ) return 1;
  const gdcm::DataElement& wave = ds.GetDataElement
      ( twave );
  if ( wave.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = wave.GetByteValue();
  assert( bv );

  std::ofstream os( outfilename );
  // Dump that to a CSV file:
  wave2stream( os, bv->GetPointer(), bv->GetLength() );
  os.close();

  return 0;
}
```

27.37 EncapsulateFileInRawData.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"

#include "magic.h" // libmagic, API to file command line tool

/*
 * Let say you want to encapsulate a file type that is not defined in DICOM
       (exe, zip, png)
 * PNG is a bad example, unless it contains transparency (which has been
       deprecated).
 * It will take care of dispatching each chunk to an appropriate data item
       (pretty much like
 * WaveformData)
 *
 * Usage:
 * ./EncapsulateFileInRawData large_input_file.exe large_input_file.dcm
 */

// TODO:
// $ file -bi /tmp/gdcm-2.1.0.pdf
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " inputfile output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  if( !gdcm::System::FileExists( filename ) ) return 1;
```

```
  size_t s = gdcm::System::FileSize(filename);

  magic_t cookie = magic_open(MAGIC_NONE);
  const char * file_type = magic_file(cookie, filename);
  magic_close(cookie);

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );

  file.GetHeader().SetDataSetTransferSyntax(
      gdcm::TransferSyntax::ImplicitVRLittleEndian
       );

  gdcm::Anonymizer anon;
  anon.SetFile( file );

  gdcm::MediaStorage ms = gdcm::MediaStorage::RawDataStorage
      ;

  gdcm::UIDGenerator gen;
  anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString()
       );
  std::cout << ms.GetString() << std::endl;
  anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate()
      );


  if (!w.Write() )
    {
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
    }

  return 0;
}
```

27.38 ExtractEncapsulatedFile.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * This example shows how one from C# context can extract a binary blob
 * and write out as a file.
 * This example is meant for pdf encapsulated file, but can be adapted for
       other type
 * of binary blob.
 *
 * DICOM file is:
 * ...
 * (0042,0010) ST (no value available)                     #   0, 0
       DocumentTitle
 * (0042,0011) OB
       25\50\44\46\2d\31\2e\32\20\0d\25\e2\e3\cf\d3\20\0d\31\30\20\30\20... # 40718, 1 EncapsulatedDocument
 * (0042,0012) LO [application/pdf]                        #  16, 1
       MIMETypeOfEncapsulatedDocument
 * ...
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ExtractEncapsulatedFile.exe some_pdf_encapsulated.dcm
 */
using System;
```

```
using gdcm;

public class ExtractEncapsulatedFile
{
  public static int Main(string[] args)
    {
    string file = args[0];
    Reader reader = new Reader();
    reader.SetFileName( file );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();
    Tag tencapsulated_stream = new Tag(0x0042,0x0011); // Encapsulated Document
    if( !ds.FindDataElement( tencapsulated_stream ) )
      {
      return 1;
      }
    // else
    DataElement de = ds.GetDataElement( tencapsulated_stream );
    ByteValue bv = de.GetByteValue();
    uint len = bv.GetLength();
    byte[] encapsulated_stream = new byte[len];
    bv.GetBuffer( encapsulated_stream, len );

    // Write out the decompressed bytes
    //System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
      System.IO.File.Open(@"/tmp/dd.pdf",
        System.IO.FileMode.Create))
      {
      System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
      writer.Write( encapsulated_stream );
      }


    return 0;
    }
}
```

27.39 ExtractEncryptedContent.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"

#include <fstream>

/*

openssl smime -encrypt -binary -aes256 -in outputfile.dcm -inform DER -out
       outputfile.der -outform DER ../trunk/Testing/Source/Data/certificate.pem

openssl smime -decrypt -binary -in out.der -inform DER -out outputfile.dcm
       -outform DER -inkey ../trunk/Testing/Source/Data/privatekey.pem ../trunk/Testing/
      Source/Data/certificate.pem

 */

int main(int argc, char *argv[])
{
  if( argc < 3 )
```

```
    {
    std::cerr << argv[0] << " input.dcm output.der" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  const gdcm::DataElement &EncryptedAttributesSequence = ds.
      GetDataElement( gdcm::Tag( 0x0400,0x0500 ) );

  gdcm::SequenceOfItems *sqi = EncryptedAttributesSequence
      .GetValueAsSQ();

  if ( !sqi || sqi->GetNumberOfItems() != 1 ) return 1;

  gdcm::Item &item = sqi->GetItem(1);

  gdcm::DataSet &nestedds = item.GetNestedDataSet(
      );

  if( ! nestedds.FindDataElement( gdcm::Tag( 0x0400,0
      x0520) ) ) return 1;

  const gdcm::DataElement &EncryptedContent = nestedds.
      GetDataElement( gdcm::Tag( 0x0400,0x0520) );

  const gdcm::ByteValue *bv = EncryptedContent.GetByteValue
      ();

  std::ofstream of( outfilename );
  of.write( bv->GetPointer(), bv->GetLength() );
  of.close();

  return 0;
}
```

27.40 ExtractIconFromFile.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to either retrieve an Icon if present somewhere
 * in the file, or else generate one.
 */
#include "gdcmImageReader.h"
#include "gdcmPNMCodec.h"
#include "gdcmIconImageFilter.h"
#include "gdcmIconImageGenerator.h"

bool WriteIconAsPNM(const char* filename, const gdcm::IconImage&
       icon)
{
  gdcm::PNMCodec pnm;
  pnm.SetDimensions( icon.GetDimensions() );
  pnm.SetPixelFormat( icon.GetPixelFormat() );
  pnm.SetPhotometricInterpretation( icon.
```

```
      GetPhotometricInterpretation() );
  pnm.SetLUT( icon.GetLUT() );
  const gdcm::DataElement& in = icon.GetDataElement
      ();
  bool b = pnm.Write( filename, in );
  assert( b );
  return true;
}

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Failed to read (or not image): " << filename << std::endl;
    return 1;
    }

  gdcm::IconImageFilter iif;
  iif.SetFile( reader.GetFile() );
  bool b = iif.Extract();

  if( b )
    {
    const gdcm::IconImage &icon = iif.GetIconImage(0
      );
    icon.Print( std::cout );

    if( !icon.GetTransferSyntax().IsEncapsulated
      () )
      {
      // Let's write out this icon as PNM file
      WriteIconAsPNM("icon.ppm", icon);
      }
    else if( icon.GetTransferSyntax() ==
      gdcm::TransferSyntax::JPEGBaselineProcess1
      || icon.GetTransferSyntax() ==
      gdcm::TransferSyntax::JPEGExtendedProcess2_4
    )
      {
      const gdcm::DataElement& in = icon.GetDataElement
      ();
      const gdcm::ByteValue *bv = in.GetByteValue();
      assert( bv );
      std::ofstream out( "icon.jpg" );
      out.write( bv->GetPointer(), bv->GetLength() );
      out.close();
      }
    }
  else
    {
    assert( iif.GetNumberOfIconImages() == 0 );
    std::cerr << "No Icon Found anywhere in file" << std::endl;

    const gdcm::Image &img = reader.GetImage();
    gdcm::IconImageGenerator iig;
    iig.AutoPixelMinMax(true);
    iig.SetPixmap( img );
    const unsigned int idims[2] = { 64, 64 };
    iig.SetOutputDimensions( idims );
    //iig.SetPixelMinMax(60, 868);
    if( !iig.Generate() ) return 1;
    const gdcm::IconImage & icon = iig.GetIconImage(
      );
    WriteIconAsPNM("icon.ppm", icon);
    }

  return 0;
}
```

27.41 Extracting_All_Resolution.cxx

```
/*=========================================================================
```

```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include <fstream>
#include <openjpeg.h>
#include <stdint.h>
#include <string.h>
#include <assert.h>
#include <gdcm_j2k.h>
#include <gdcm_jp2.h>
#include<iostream>
#include <cstring>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmSystem.h"
#include <fstream>



#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"

void error_callback(const char *msg, void *) {
  (void)msg;
}
void warning_callback(const char *msg, void *) {
  (void)msg;
}
void info_callback(const char *msg, void *) {
  (void)msg;
}


bool Write_Resolution(gdcm::StreamImageWriter &
      theStreamWriter, const char *filename, int res, std::ostream& of, int flag,
      gdcm::SequenceOfItems *sq, int No_Of_Resolutions)
{
  std::ifstream is;
  is.open( filename );
  opj_dparameters_t parameters;  /* decompression parameters */
  opj_event_mgr_t event_mgr;    /* event manager */
  opj_dinfo_t* dinfo;  /* handle to a decompressor */
  opj_cio_t *cio;
  opj_image_t *image = NULL;
  // FIXME: Do some stupid work:
  is.seekg( 0, std::ios::end);
  std::streampos buf_size = is.tellg();
  char *dummy_buffer = new char[(unsigned int)buf_size];
  is.seekg(0, std::ios::beg);
  is.read( dummy_buffer, buf_size);
  unsigned char *src = (unsigned char*)dummy_buffer;
  uint32_t file_length = (uint32_t)buf_size; // 32bits truncation should be ok
       since DICOM cannot have larger than 2Gb image


  /* configure the event callbacks (not required) */
```

```
  memset(&event_mgr, 0, sizeof(opj_event_mgr_t));
  event_mgr.error_handler = error_callback;
  event_mgr.warning_handler = warning_callback;
  event_mgr.info_handler = info_callback;

  /* set decoding parameters to default values */
  opj_set_default_decoder_parameters(&parameters);

  // default blindly copied
  parameters.cp_layer=0;
  parameters.cp_reduce= res;
  //   parameters.decod_format=-1;
  //   parameters.cod_format=-1;

  const char jp2magic[] = "\x00\x00\x00\x0C\x6A\x50\x20\x20\x0D\x0A\x87\x0A";
  if( memcmp( src, jp2magic, sizeof(jp2magic) ) == 0 )
    {
    /* JPEG-2000 compressed image data ... sigh */
    // gdcmData/ELSCINT1_JP2vsJ2K.dcm
    // gdcmData/MAROTECH_CT_JP2Lossy.dcm
    //gdcmWarningMacro( "J2K start like JPEG-2000 compressed image data instead
       of codestream" );
    parameters.decod_format = 1; //JP2_CFMT;
    //assert(parameters.decod_format == JP2_CFMT);
    }
  else
    {
    /* JPEG-2000 codestream */
    //parameters.decod_format = J2K_CFMT;
    //assert(parameters.decod_format == J2K_CFMT);
    assert( 0 );
    }
  parameters.cod_format = 11; // PGX_DFMT;
  //assert(parameters.cod_format == PGX_DFMT);

  /* get a decoder handle */
    dinfo = opj_create_decompress(CODEC_JP2);

  /* catch events using our callbacks and give a local context */
  opj_set_event_mgr((opj_common_ptr)dinfo, &event_mgr, NULL);

  /* setup the decoder decoding parameters using user parameters */
  opj_setup_decoder(dinfo, &parameters);

  /* open a byte stream */
  cio = opj_cio_open((opj_common_ptr)dinfo, src, file_length);

  /* decode the stream and fill the image structure */
  image = opj_decode(dinfo, cio);
  if(!image) {
    opj_destroy_decompress(dinfo);
    opj_cio_close(cio);
    //gdcmErrorMacro( "opj_decode failed" );
    return 1;
  }

      opj_cp_t * cp = ((opj_jp2_t*)dinfo->jp2_handle)->j2k->cp;
      opj_tcp_t *tcp = &cp->tcps[0];
      opj_tccp_t *tccp = &tcp->tccps[0];
    /*   std::cout << "\n No of Cols In Image" << image->x1;
      std::cout << "\n No of Rows In Image" << image->y1;
      std::cout << "\n No of Components in Image" << image->numcomps;
      std::cout << "\n No of Resolutions"<< tccp->numresolutions << "\n";
*/
       opj_j2k_t* j2k = NULL;
       opj_jp2_t* jp2 = NULL;
       jp2 = (opj_jp2_t*)dinfo->jp2_handle;
       int reversible = jp2->j2k->cp->tcps->tccps->qmfbid;
       //std:: cout << reversible;
      int compno = 0;
      opj_image_comp_t *comp = &image->comps[compno];
      int Dimensions[2];
      Dimensions[0]= comp->w;
      Dimensions[1] = comp->h;
      opj_cio_close(cio);
      unsigned long len = Dimensions[0]*Dimensions[1] * image->numcomps;
      //std::cout << "\nTest" <<image->comps[0].factor;
      char *raw = new char[len];
  for (unsigned int compno = 0; compno < (unsigned int)image->numcomps; compno
     ++)
    {
```

```
    opj_image_comp_t *comp = &image->comps[compno];

    int w = image->comps[compno].w;
    int h = image->comps[compno].h;
      uint8_t *data8 = (uint8_t*)raw + compno;
      for (int i = 0; i < w * h ; i++)
        {
        int v = image->comps[compno].data[i];
        *data8 = (uint8_t)v;
        data8 += image->numcomps;
        }
    }

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  file.GetHeader().SetDataSetTransferSyntax(
      gdcm::TransferSyntax::ExplicitVRLittleEndian
       );

  gdcm::UIDGenerator uid;
  gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP
       Instance UID
  de.SetVR( gdcm::VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );

  gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
  de1.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::CTImageStorage
       );
  de1.SetByteValue( ms.GetString(), strlen(ms.GetString()));
  ds.Insert( de1 );

  const char mystr[] = "MONOCHROME2 ";
  gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  de2.SetVR( gdcm::VR::CS );
  de2.SetByteValue(mystr, strlen(mystr));
  ds.Insert( de2 );

   gdcm::Attribute<0x0028,0x0010> row = {image->
      comps[0].w};
   //row.SetValue(512);
   ds.Insert( row.GetAsDataElement() );
//  w.SetCheckFileMetaInformation( true );
   gdcm::Attribute<0x0028,0x0011> col = {image->
      comps[0].h};
   ds.Insert( col.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0008>
      Number_Of_Frames = {1};
   ds.Insert( Number_Of_Frames.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0100> at = {8};
   ds.Insert( at.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0002> at1 = {image->
      numcomps};
   ds.Insert( at1.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0101> at2 = {8};
   ds.Insert( at2.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0102> at3 = {7};
  ds.Insert( at3.GetAsDataElement() );



  if (flag == 1)
  {

    for (int i=0; i < No_Of_Resolutions; i++)
    {

     int a = 1;
     int b =1;

     while(a!=((No_Of_Resolutions)-i))
        {
```

```
        b = b*2;
        a = a+1;
       }
    uint16_t row = (image->y1)/b;
    uint16_t col = (image->x1)/b;
    //std::cout << row;
    gdcm::Element<gdcm::VR::IS,gdcm::VM::VM1>
      el2;
    el2.SetValue(i+1);
    gdcm::DataElement rfn = el2.GetAsDataElement
     ();     //ulr --> upper left row
    rfn.SetTag( gdcm::Tag(0x0008,0x1160) );

    gdcm::Element<gdcm::VR::US,gdcm::VM::VM2>
      el;
    el.SetValue(1,0);
    el.SetValue(1,1);
    gdcm::DataElement ulr = el.GetAsDataElement
     ();     //ulr --> upper left col/row
    ulr.SetTag( gdcm::Tag(0x0048,0x0201) );

    gdcm::Element<gdcm::VR::US,gdcm::VM::VM2>
      el1;
    el1.SetValue(col,0);
    el1.SetValue(row,1);
    gdcm::DataElement brr = el1.GetAsDataElement
     ();
    brr.SetTag( gdcm::Tag(0x0048,0x0202) );            //brr
      --> bottom right col/row
   gdcm::Item it;
   gdcm::DataSet &nds = it.GetNestedDataSet();
   nds.Insert( rfn );
   nds.Insert(ulr);
   nds.Insert(brr);

   sq->AddItem(it);
  }

  gdcm::Writer w1;
  gdcm::File &file1 = w1.GetFile();
  gdcm::DataSet &ds1 = file1.GetDataSet();
  file1.GetHeader().SetDataSetTransferSyntax(
      gdcm::TransferSyntax::ExplicitVRLittleEndian
       );

  gdcm::UIDGenerator uid1;
  gdcm::DataElement dea( gdcm::Tag(0x8,0x18) ); //
       SOP Instance UID
  dea.SetVR( gdcm::VR::UI );
  const char *u1 = uid1.Generate();
  dea.SetByteValue( u1, strlen(u1) );
  ds1.Insert( dea );

  gdcm::DataElement deb( gdcm::Tag(0x8,0x16) );
  deb.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms1(
      gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
       );
  deb.SetByteValue( ms1.GetString(), strlen(ms1.GetString()));
  ds1.Insert( deb );

  const char mystr1[] = "MONOCHROME2 ";
  gdcm::DataElement dec( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  dec.SetVR( gdcm::VR::CS );
  dec.SetByteValue(mystr, strlen(mystr1));
  ds1.Insert( dec );

   gdcm::Attribute<0x0028,0x0010> row1 = {image->
      y1};
   //row.SetValue(512);
   ds1.Insert( row1.GetAsDataElement() );
//  w.SetCheckFileMetaInformation( true );
   gdcm::Attribute<0x0028,0x0011> col1 = {image->
      x1};
   ds1.Insert( col1.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0008>
      Number_Of_Frames1 = {tccp->numresolutions};
   ds1.Insert( Number_Of_Frames1.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0100> ata = {8};
```

```
   ds1.Insert( ata.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0002> atb = {image->
      numcomps};
   ds1.Insert( atb.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0101> atc = {8};
   ds1.Insert( atc.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0102> atd = {7};
  ds1.Insert( atd.GetAsDataElement() );

   theStreamWriter.SetFile(file1);

  gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
  des.SetVR(gdcm::VR::SQ);
  //des.SetVR(gdcm::VM::VM1);
  des.SetValue(*sq);
  des.SetVLToUndefined();

  ds1.Insert(des);


  if (!theStreamWriter.WriteImageInformation()){
      std::cerr << "unable to write image information" << std::endl;
      return 1; //the CanWrite function should prevent getting here, else,
      //that's a test failure
    }

 }

   theStreamWriter.SetFile(file);

  if (!theStreamWriter.CanWriteFile()){
      delete [] raw;
      std::cout << "Not able to write";
      return 0;//this means that the file was unwritable, period.
      //very similar to a ReadImageInformation failure
    }
else
   std::cout<<"\nabletoread";

 // Important to write here
 std::vector<unsigned int> extent = gdcm::ImageHelper::GetDimensionsValue
      (file);

    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 4;
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of
       theChunkSize
    unsigned short zmax = extent[2];
    std::cout << "\n"<<xmax << "\n" << ymax<<"\n"<<zmax<<"\n" << image->
      numcomps<<"\n";


    if (xmax == 0 || ymax == 0)
      {
      std::cerr << "Image has no size, unable to write zero-sized image." <<
      std::endl;
      return 0;
      }


    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure
       to grab
    //the bytes sequentially.  So, store how far you got in the buffer with
       each iteration.
    for (z = 0; z < zmax; ++z){
      for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z
      , z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength
      ();
        std::cout << "\n" <<len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(raw[prevLen]), len);
        std::cout << "\nable to write";
```

```
        if (!theStreamWriter.Write(finalBuffer, len)){
          std::cerr << "writing failure:" << "output.dcm" << " at y = " << y <<
       " and z= " << z << std::endl;
          delete [] raw;
          delete [] finalBuffer;
          return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
      }
    }
    delete raw;

  delete[] src;  //FIXME

if(dinfo) {
    opj_destroy_decompress(dinfo);
  }

 opj_image_destroy(image);

 return true;

}

bool Different_Resolution( gdcm::StreamImageWriter &
      theStreamWriter, const char *filename, int res, std::ostream& of)
{
  //std::vector<std::string>::const_iterator it = filenames.begin();
  bool b = true;
  int flag = 1;

 gdcm::SmartPointer<gdcm::SequenceOfItems>
       sq = new gdcm::SequenceOfItems();
 sq->SetLengthToUndefined();


  for(int i = res-1 ; i>=0; --i)
  {
    b = b && Write_Resolution( theStreamWriter, filename, i, of ,flag,sq,res);
   // b = b && Get_Resolution( theStreamWriter, filename, i, of ,0);
    flag = 0;
  }
  //b = b && Get_Lowest_Resolution( writer, sq, filename, res-1 );
  //b = b && PopulateSingeFile( writer, sq, jpeg, filename2 );
  //image.SetDimension(2,  res )
  return b;
}


int main(int argc, char *argv[])
{


   if( argc < 4 )
    {
    std::cerr << argv[0] << " input.jp2 output.dcm No. Of Resolutions " <<
      std::endl;
    return 1;
    }
   const char *filename = argv[1];
  const char *outfilename = argv[2];
  char *resolutions = argv[3];
   int res = int((*resolutions)-48);
    //std:: cout << "\nres"<< res;
   gdcm::StreamImageWriter theStreamWriter;

   std::ofstream of;
   of.open( outfilename, std::ios::out | std::ios::binary );
   theStreamWriter.SetStream(of);

   if( !Different_Resolution( theStreamWriter, filename,res,of ) ) return 1;

 uint16_t firstTag1 =  0xfffe;
 uint16_t secondTag1 = 0xe0dd;
 uint32_t thirdTag1 =  0x00000000;
 //uint16_t fourthTag1 = 0xffff;
 const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
 char* tmpBuffer2 = new char[theBufferSize1];
 memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
```

```
 memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
 //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
 assert( of && !of.eof() && of.good() );
 of.write(tmpBuffer2, theBufferSize1);
 of.flush();
 assert( of );


return 0;
}
```

27.42 Fake_Image_Using_Stream_Image_Writer.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcmReader.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"

int main(int, char *[])
{

  char * buffer = new char[ 256 * 256 *3 ];
  // *p = (uint8_t*)buffer;
  char * p = buffer;

  gdcm::Trace::DebugOn();
  gdcm::Trace::WarningOn();

  for(int row = 0; row < 256; ++row)
  {
    for(int col = 0; col < 256; ++col)
      //for(int b = 0; b < 256; ++b)
      {
        *p++ = 255;
        *p++ = 0;
        *p++ = 0;
       }
  }

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  file.GetHeader().SetDataSetTransferSyntax(
      gdcm::TransferSyntax::ExplicitVRLittleEndian
       );

  gdcm::UIDGenerator uid;
  gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP
       Instance UID
  de.SetVR( gdcm::VR::UI );
```

```
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );

  gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
  de1.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms(
      gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
       );
  de1.SetByteValue( ms.GetString(), strlen(ms.GetString
      ()));
  ds.Insert( de1 );

  const char mystr[] = "RGB";
  gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  de2.SetVR( gdcm::VR::CS );
  de2.SetByteValue(mystr, strlen(mystr));
  ds.Insert( de2 );

   gdcm::Attribute<0x0028,0x0010> row = {256};
   //row.SetValue(512);
   ds.Insert( row.GetAsDataElement() );
 //  w.SetCheckFileMetaInformation( true );
   gdcm::Attribute<0x0028,0x0011> col = {256};
   ds.Insert( col.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0008>
      Number_Of_Frames = {1};
   ds.Insert( Number_Of_Frames.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0100> at = {8};
   ds.Insert( at.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0002> at1 = {3}; //
      bits per pixel
   ds.Insert( at1.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0101> at2 = {8};
   ds.Insert( at2.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0102> at3 = {7};
   ds.Insert( at3.GetAsDataElement() );

 gdcm::Attribute<0x0028,0x006> at4 = {0};
 ds.Insert( at4.GetAsDataElement() );

 gdcm::Attribute<0x0028,0x0103> at5 = {0};
 ds.Insert( at5.GetAsDataElement() );

 //de.SetTag(gdcm::Tag(0x7fe0,0x0010));
 //ds.Insert(de);

  gdcm::StreamImageWriter theStreamWriter;
gdcm::SmartPointer<gdcm::SequenceOfItems>
       sq = new gdcm::SequenceOfItems();
 sq->SetLengthToUndefined();

     uint16_t row1 = 256;
     uint16_t col1 = 256;
     //std::cout << row;

     gdcm::Element<gdcm::VR::IS,gdcm::VM::VM1>
       el2;
     el2.SetValue(1);
     gdcm::DataElement rfn = el2.GetAsDataElement
      ();     //rfn ---> reference frame number
     rfn.SetTag( gdcm::Tag(0x0008,0x1160) );

     gdcm::Element<gdcm::VR::US,gdcm::VM::VM2>
       el;
     el.SetValue(1,0);
     el.SetValue(1,1);
     gdcm::DataElement ulr = el.GetAsDataElement
      ();     //ulr --> upper left col/row
     ulr.SetTag( gdcm::Tag(0x0048,0x0201) );

     gdcm::Element<gdcm::VR::US,gdcm::VM::VM2>
       el1;
     el1.SetValue(col1,0);
     el1.SetValue(row1,1);
```

```
     gdcm::DataElement brr = el1.GetAsDataElement
       ();
     brr.SetTag( gdcm::Tag(0x0048,0x0202) );            //brr
       --> bottom right col/row

    gdcm::Item it;
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert( rfn );
    nds.Insert(ulr);
    nds.Insert(brr);

    sq->AddItem(it);

  gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();

  ds.Insert(des);


 theStreamWriter.SetFile(file);

  std::ofstream of;
  of.open( "output.dcm", std::ios::out | std::ios::binary );
  theStreamWriter.SetStream(of);


if (!theStreamWriter.CanWriteFile()){
      delete [] buffer;
      std::cout << "Not able to write";
      return 0;//this means that the file was unwritable, period.
      //very similar to a ReadImageInformation failure
    }
else
   std::cout<<"\nabletoread";

if (!theStreamWriter.WriteImageInformation()){
      std::cerr << "unable to write image information" << std::endl;
      delete [] buffer;
      return 1; //the CanWrite function should prevent getting here, else,
      //that's a test failure
    }

 std::vector<unsigned int> extent =
      gdcm::ImageHelper::GetDimensionsValue
      (file);

    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 1;
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of
       theChunkSize
    unsigned short zmax = extent[2];

    std::cout << xmax << ymax << zmax;

    if (xmax == 0 || ymax == 0)
      {
      std::cerr << "Image has no size, unable to write zero-sized image." <<
      std::endl;
      return 0;
      }

    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure
       to grab
    //the bytes sequentially.  So, store how far you got in the buffer with
       each iteration.
    for (z = 0; z < zmax; ++z){
      for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z
      , z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength
      ();
        std::cout << "\n" <<len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(buffer[prevLen]), len);
        std::cout << "\nable to write";
```

```
        if (!theStreamWriter.Write(finalBuffer, len)){
          std::cerr << "writing failure:" << "output.dcm" << " at y = " << y <<
       " and z= " << z << std::endl;
          delete [] buffer;
          delete [] finalBuffer;
          return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
      }
    }
    delete buffer;

 uint16_t firstTag1 =  0xfffe;
 uint16_t secondTag1 = 0xe0dd;
 uint32_t thirdTag1 =  0x00000000;
 //uint16_t fourthTag1 = 0xffff;
 const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
 char* tmpBuffer2 = new char[theBufferSize1];
 memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
 //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
 assert( of && !of.eof() && of.good() );
 of.write(tmpBuffer2, theBufferSize1);
 of.flush();
 assert( of );


  return 0;
}
```

27.43 FindAllPatientName.py

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################
"""
This example shows how one can use the gdcm.CompositeNetworkFunctions class
for executing a C-FIND query
It will print the list of patient name found

Usage:

 python FindAllPatientName.py

"""

import gdcm

# Patient Name
tag = gdcm.Tag(0x10,0x10)
de = gdcm.DataElement(tag)

# Search all patient name where string match 'F*'
de.SetByteValue('F*',gdcm.VL(2))

ds = gdcm.DataSet()
ds.Insert(de)

cnf = gdcm.CompositeNetworkFunctions()
theQuery = cnf.ConstructQuery (gdcm.ePatientRootType,gdcm.ePatient,ds)

#print theQuery.ValidateQuery()

# prepare the variable for output
ret = gdcm.DataSetArrayType()
```

```
# Execute the C-FIND query
cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')

for i in range(0,ret.size()):
  print "Patient #",i
  print ret[i]
```

27.44 FixBrokenJ2K.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmFile.h"

// http://www.lost.in.ua/dicom/c.dcm
//
// -> BuggyJ2Kvvvua-fixed2-j2k.dcm

/*
 * This program attemps to fix a broken J2K/DICOM:
 * It contains 2 bugs:
 * 1. The first 8 bytes seems to be random bytes: remove them
 * 2. YCC is set to 1, while image is grayscale need to set it back to 0
 *
 * Ref:
 * It's a software from http://rentgenprom.ru/ , shipped with universal digital
       radiographic units
 * "ProScan-2000". The Ukrainian manufacturer developed own digital
       radiographic unit and it is
 * compatible with software from "ProScan-2000".
 */
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  const gdcm::DataElement &pixeldata = file.GetDataSet
      ().GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
  const gdcm::SequenceOfFragments *sqf = pixeldata.
      GetSequenceOfFragments();
  if( !sqf )
    {
    return 1;
    }
  const gdcm::Fragment &frag = sqf->GetFragment(0);

  const gdcm::ByteValue *bv = frag.GetByteValue();
  const char *ptr = bv->GetPointer();
  size_t len = bv->GetLength();
```

```
  const char sig[] = "\x00\x00\x00\x00\x6A\x70\x32\x63";
  if( memcmp(ptr, sig, sizeof(sig) != 0 ) )
    {
    std::cerr << "magic random signature not found" << std::endl;
    return 1;
    }

  // Apparently the flag to enable a color transform on 3 color components is
       set in
  // the COD marker. (YCC is byte[6] in the COD marker)
  // we need to disable this flag;
  const char *cod_marker = ptr + 0x35; /* 0x2d + 0x8 */ // FIXME
  if( cod_marker[0] == (char)0xff && cod_marker[1] == 0x52 )
    {
    // found start of COD
    if( cod_marker[6+2] == 1 )
      {
      // Change in place:
      *((char*)cod_marker + 6+2) = 0;
      // Prepare a new DataElement:
      gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0
      x0010) );
      pixeldata.SetVR( gdcm::VR::OB );
      gdcm::SmartPointer<gdcm::SequenceOfFragments>
       sq = new gdcm::SequenceOfFragments;

      gdcm::Fragment frag;
      // remove 8 first bytes:
      frag.SetByteValue( ptr + 8, len - 8 );
      sq->AddFragment( frag );
      pixeldata.SetValue( *sq );
      file.GetDataSet().Replace( pixeldata );
      }
    else
      {
      return 1;
      }
    }
  else
    {
    std::cerr << "COD not found" << (int)cod_marker[0] << std::endl;
    return 1;
    }

  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename );
  writer.CheckFileMetaInformationOff();
  if( !writer.Write() )
    {
    std::cerr << "Could not write" << std::endl;
    }

  // paranoid check:
  gdcm::ImageReader ireader;
  ireader.SetFileName( outfilename );
  if( !ireader.Read() )
    {
    std::cerr << "file written is still not valid, please report" << std::endl;
    return 1;
    }


  return 0;
}
```

27.45 FixCommaBug.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
```

```
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

"""
Using LC_NUMERIC set to something not compatible with "C" it is possible to
       write out "," instead of
"." as required by the DICOM standard
Issue is still current (IMHO) with gdcm 2.0.9
"""

import gdcm
import sys

filename = sys.argv[1]
outname = sys.argv[2]

# read
r = gdcm.Reader()
r.SetFileName( filename )
if not r.Read():
  print "not valid"
  sys.exit(1)

file = r.GetFile()
dataset = file.GetDataSet()

ano = gdcm.Anonymizer()
ano.SetFile( file )

tags = [
gdcm.Tag(0x0018,0x1164),
gdcm.Tag(0x0018,0x0088),
gdcm.Tag(0x0018,0x0050),
gdcm.Tag(0x0028,0x0030),
]

for tag in tags:
  print tag
  if dataset.FindDataElement( tag ):
    pixelspacing = dataset.GetDataElement( tag )
    #print pixelspacing
    bv = pixelspacing.GetByteValue()
    str = bv.GetBuffer()
    #print bv.GetLength()
    #print len(str)
    new_str = str.replace(",",".")
    # Need to explicitly pass bv.GetLength() to remove any trailing garbage
    ano.Replace( tag, new_str, bv.GetLength() )

#print dataset

w = gdcm.Writer()
w.SetFile( file )
w.SetFileName( outname )
if not w.Write():
  print "Cannot write"
  sys.exit(1)

# paranoid:
image_reader = gdcm.ImageReader()
image_reader.SetFileName( outname )
if not image_reader.Read():
  print "there is still a comma"
  sys.exit(1)

print "Sucess!"
sys.exit(0) # success
```

27.46 FixJAIBugJPEGLS.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"

#include <fstream>

#include "gdcm_charls.h"

/*
 * This small example should show how one can handle the famous JAI-JPEGLS bug
 * It will take in as invalid DICOM/JAI-JPEG-LS and write out as Explicit
       Little
 * Endian. One can use `gdcmconv --jpegls` to recompress properly
 *
 * References:
 * http://charls.codeplex.com/discussions/230307?ProjectName=charls
 * http://charls.codeplex.com/workitem/7297
 * http://www.dcm4che.org/jira/browse/DCM-442
 * http://www.dcm4che.org/jira/browse/DCMEE-1144
 * http://java.net/jira/browse/JAI_IMAGEIO_CORE-183
 *
 * Explanation of the issue:
 *
 * Seems, the error is in the calculation of the default values for thresholds
       T1,
 * T2, T3, in particular min(MAXVAL, 4095) is not applied in
 *
 * FACTOR = (min(MAXVAL, 4095) + 128)/256
 *
 * as specified in http://www.itu.int/rec/T-REC-T.87-199806-I/en .
 *
 */
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::FileMetaInformation::SetSourceApplicationEntityTitle
      ( "FixJAIBugJPEGLS" );

  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::Image &image = reader.GetImage();
  //unsigned long len = image.GetBufferLength();
  const gdcm::DataElement & in =
    reader.GetFile().GetDataSet().GetDataElement
      ( gdcm::Tag(0x7fe0,0x0010) );
  const gdcm::SequenceOfFragments *sf = in.
      GetSequenceOfFragments();
  if( !sf )
    {
    std::cerr << "No pixel data (or not encapsulated)" << std::endl;
    return 1;
    }
  const unsigned int *dims = image.GetDimensions();
  if ( sf->GetNumberOfFragments() != dims[2] )
    {
    std::cerr << "Unsupported" << std::endl;
    return 1;
    }

//  unsigned long totalLen = sf->ComputeByteLength();
  std::vector<BYTE> rgbyteOutall;
  for(unsigned int i = 0; i < sf->GetNumberOfFragments(); +
```

```
      +i)
    {
    const gdcm::Fragment &frag = sf->GetFragment(i);
    if( frag.IsEmpty() ) return 1;
    const gdcm::ByteValue *bv = frag.GetByteValue();
    if( !bv ) return 1;
    unsigned long totalLen = bv->GetLength();

    std::vector<char> vbuffer;
    vbuffer.resize( totalLen );
    char *buffer = &vbuffer[0];
    bv->GetBuffer(buffer, totalLen);
    const BYTE* pbyteCompressed0 = (const BYTE*)buffer;
    while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
      {
      totalLen--;
      }

#ifdef GDCM_USE_SYSTEM_CHARLS
    JlsParameters metadata;
#else
    JlsParamaters metadata;
#endif
    if (JpegLsReadHeader(buffer, totalLen, &metadata) != OK)
      {
      std::cerr << "Cant parse jpegls" << std::endl;
      return false;
      }

    std::cout << metadata.width << std::endl;
    std::cout << metadata.height << std::endl;
    std::cout << metadata.bitspersample << std::endl;

    gdcm::PixelFormat const & pf = image.GetPixelFormat
      ();
    std::cout << pf << std::endl;

    // http://charls.codeplex.com/discussions/230307?ProjectName=charls
    unsigned char marker_lse_13[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0x1F, 0xFF,
      0x00, 0x22,  // T1 = 34
      0x00, 0x83,  // T2 = 131
      0x02, 0x24,  // T3 = 548
      0x00, 0x40
    };

    unsigned char marker_lse_14[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0x3F, 0xFF,
      0x00, 0x42, // T1 = 66
      0x01, 0x03, // T2 = 259
      0x04, 0x44, // T3 = 1092
      0x00, 0x40
    };

    unsigned char marker_lse_15[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0x7F, 0xFF,
      0x00, 0x82, // T1 = 130
      0x02, 0x03, // T2 = 515
      0x08, 0x84, // T3 = 2180
      0x00, 0x40
    };

    unsigned char marker_lse_16[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0xFF, 0xFF,
      0x01, 0x02, // T1 = 258
      0x04, 0x03, // T2 = 1027
      0x11, 0x04, // T3 = 4356
      0x00, 0x40
    };

    const unsigned char *marker_lse = NULL;
    switch( metadata.bitspersample )
      {
```

```
    case 13:
      marker_lse = marker_lse_13;
      break;
    case 14:
      marker_lse = marker_lse_14;
      break;
    case 15:
      marker_lse = marker_lse_15;
      break;
    case 16:
      marker_lse = marker_lse_16;
      break;
      }
    if( !marker_lse )
      {
      std::cerr << "Cant handle: " << metadata.bitspersample << std::endl;
      return 1;
      }

    // FIXME: One should recompute the value for 0x0F
    vbuffer.insert (vbuffer.begin() + 0x0F, marker_lse, marker_lse+15);

#if 0
    std::ofstream of( "/tmp/d.jls" );
    of.write( &vbuffer[0], vbuffer.size() );
    of.close();
#endif

    const char *pbyteCompressed = &vbuffer[0];
    unsigned int cbyteCompressed = vbuffer.size(); // updated legnth

#ifdef GDCM_USE_SYSTEM_CHARLS
    JlsParameters params = {0};
#else
    JlsParamaters params = {0};
#endif
    JpegLsReadHeader(pbyteCompressed, cbyteCompressed, &params);

    std::vector<BYTE> rgbyteOut;
    //rgbyteOut.resize( image.GetBufferLength() );
    rgbyteOut.resize(params.height *params.width * ((params.bitspersample + 7)
        / 8) * params.components);

#ifdef GDCM_USE_SYSTEM_CHARLS
    JLS_ERROR result =
      JpegLsDecode(&rgbyteOut[0], rgbyteOut.size(), pbyteCompressed,
      cbyteCompressed, &params );
#else
    JLS_ERROR result =
      JpegLsDecode(&rgbyteOut[0], rgbyteOut.size(), pbyteCompressed,
      cbyteCompressed );
#endif
    if (result != OK)
      {
      std::cerr << "Could not patch JAI-JPEGLS" << std::endl;
      return 1;
      }
    rgbyteOutall.insert( rgbyteOutall.end(), rgbyteOut.begin(), rgbyteOut.end()
       );
    }

  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0
      x0010) );
  pixeldata.SetVR( gdcm::VR::OW );
  pixeldata.SetByteValue( (char*)&rgbyteOutall[0], rgbyteOutall.
      size() );


  // Add the pixel data element
  reader.GetFile().GetDataSet().Replace( pixeldata );
  reader.GetFile().GetHeader().SetDataSetTransferSyntax
      (
    gdcm::TransferSyntax::ExplicitVRLittleEndian
      );

  gdcm::Writer writer;
  writer.SetFileName( outfilename );
  writer.SetFile( reader.GetFile() );
  writer.Write();

  std::cout << "Success !" << std::endl;
```

```
  return 0;
}
```

27.47 gdcmorthoplanes.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#include "vtkActor.h"
#include "vtkCamera.h"
#include "vtkMatrix4x4.h"
#include "vtkTransform.h"
#include "vtkAssembly.h"
#include "vtkCellPicker.h"
#include "vtkCommand.h"
#include "vtkImageActor.h"
#include "vtkImageMapToColors.h"
#include "vtkImageOrthoPlanes.h"
#include "vtkImagePlaneWidget.h"
#include "vtkImageReader.h"
#include "vtkInteractorEventRecorder.h"
#include "vtkLookupTable.h"
#include "vtkOutlineFilter.h"
#include "vtkPolyDataMapper.h"
#include "vtkProperty.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderer.h"
#include "vtkVolume16Reader.h"
#include "vtkImageData.h"
#include "vtkImageChangeInformation.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkAxesActor.h"
#include "vtkCaptionActor2D.h"
#include "vtkTextProperty.h"
#include "vtkPropAssembly.h"

#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkStringArray.h"

#include "gdcmSystem.h"
#include "gdcmDirectory.h"
#include "gdcmIPPSorter.h"

#ifndef vtkFloatingPointType
#define vtkFloatingPointType float
#endif

//----------------------------------------------------------------------------
class vtkOrthoPlanesCallback : public vtkCommand
{
public:
  static vtkOrthoPlanesCallback *New()
  { return new vtkOrthoPlanesCallback; }

  void Execute( vtkObject *caller, unsigned long vtkNotUsed( event ),
                void *callData )
  {
    vtkImagePlaneWidget* self =
      reinterpret_cast< vtkImagePlaneWidget* >( caller );
    if(!self) return;

    double* wl = static_cast<double*>( callData );
```

```
    if ( self == this->WidgetX )
      {
      this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
      }
    else if( self == this->WidgetY )
      {
      this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
      }
    else if (self == this->WidgetZ)
      {
      this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
      }
  }

  vtkOrthoPlanesCallback():WidgetX( 0 ), WidgetY( 0 ), WidgetZ ( 0 ) {}

  vtkImagePlaneWidget* WidgetX;
  vtkImagePlaneWidget* WidgetY;
  vtkImagePlaneWidget* WidgetZ;
};

int main( int argc, char *argv[] )
{
  //char* fname = vtkTestUtilities::ExpandDataFileName(argc, argv,
       "Data/headsq/quarter");

  //vtkVolume16Reader* v16 =  vtkVolume16Reader::New();
  //  v16->SetDataDimensions( 64, 64);
  //  v16->SetDataByteOrderToLittleEndian();
  //  v16->SetImageRange( 1, 93);
  //  v16->SetDataSpacing( 3.2, 3.2, 1.5);
  //  v16->SetFilePrefix( fname );
  //  v16->SetDataMask( 0x7fff);
  //  v16->Update();
  std::vector<std::string> filenames;
  if( argc < 2 )
    {
    std::cerr << argv[0] << " filename1.dcm [filename2.dcm ...]\n";
    return 1;
    }
  else
    {
    // Is it a single directory ? If so loop over all files contained in it:
    const char *filename = argv[1];
    if( argc == 2 && gdcm::System::FileIsDirectory
      ( filename ) )
      {
      std::cout << "Loading directory: " << filename << std::endl;
      bool recursive = false;
      gdcm::Directory d;
      d.Load(filename, recursive);
      gdcm::Directory::FilenamesType const &files
       = d.GetFilenames();
      for( gdcm::Directory::FilenamesType::const_iterator it = files.begin();
      it != files.end(); ++it )
        {
        filenames.push_back( it->c_str() );
        }
      }
    else // list of files passed directly on the cmd line:
        // discard non-existing or directory
      {
      for(int i=1; i < argc; ++i)
        {
        filename = argv[i];
        if( gdcm::System::FileExists( filename ) )
          {
          if( gdcm::System::FileIsDirectory(
      filename ) )
            {
            std::cerr << "Discarding directory: " << filename << std::endl;
            }
          else
            {
            filenames.push_back( filename );
            }
          }
```

```
        else
          {
          std::cerr << "Discarding non existing file: " << filename <<
      std::endl;
          }
        }
      }
    //names->Print( std::cout );
    }

  //gdcm::Trace::DebugOn();
  //gdcm::Trace::WarningOn();
  gdcm::IPPSorter s;
  s.SetComputeZSpacing( true );
  s.SetZSpacingTolerance( 1e-3 );
  bool b = s.Sort( filenames );
  if( !b )
    {
    std::cerr << "Failed to sort files" << std::endl;
    return 1;
    }
  std::cout << "Sorting succeeded:" << std::endl;
  s.Print( std::cout );

  std::cout << "Found z-spacing:" << std::endl;
  std::cout << s.GetZSpacing() << std::endl;
  double ippzspacing = s.GetZSpacing();

  const std::vector<std::string> & sorted = s.GetFilenames();
  vtkStringArray *files = vtkStringArray::New();
  std::vector< std::string >::const_iterator it = sorted.begin();
  for( ; it != sorted.end(); ++it)
    {
    const std::string &f = *it;
    files->InsertNextValue( f.c_str() );
    }

  //delete[] fname;
  vtkGDCMImageReader * reader = vtkGDCMImageReader::New
      ();
  //reader->SetFileLowerLeft( 1 );
  reader->SetFileNames( files );
  reader->Update(); // important
  //reader->GetOutput()->Print( std::cout );
  //vtkFloatingPointType range[2];
  //reader->GetOutput()->GetScalarRange(range);
  //std::cout << "Range: " << range[0] << " " << range[1] << std::endl;

  const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();

  vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
  v16->SetInput( reader->GetOutput() );
  v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  v16->Update();

#if 0
    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
      ();
    writer->SetInput( v16->GetOutput() );
    writer->SetFileLowerLeft( reader->GetFileLowerLeft() );
    writer->SetDirectionCosines( reader->GetDirectionCosines
      () );
    writer->SetImageFormat( reader->GetImageFormat() );
    writer->SetFileDimensionality( 3); //reader->GetFileDimensionality() );
    writer->SetMedicalImageProperties( reader->
      GetMedicalImageProperties() );
    writer->SetShift( reader->GetShift() );
    writer->SetScale( reader->GetScale() );
    writer->SetFileName( "out.dcm" );
    writer->Write();
#endif


  files->Delete();

  vtkOutlineFilter* outline = vtkOutlineFilter::New();
    outline->SetInputConnection(v16->GetOutputPort());

  vtkPolyDataMapper* outlineMapper = vtkPolyDataMapper::New();
    outlineMapper->SetInputConnection(outline->GetOutputPort());
```

```
vtkActor* outlineActor =  vtkActor::New();
  outlineActor->SetMapper( outlineMapper);

vtkRenderer* ren1 = vtkRenderer::New();
vtkRenderer* ren2 = vtkRenderer::New();

vtkRenderWindow* renWin = vtkRenderWindow::New();
  renWin->AddRenderer(ren2);
  renWin->AddRenderer(ren1);

vtkRenderWindowInteractor* iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renWin);

vtkCellPicker* picker = vtkCellPicker::New();
  picker->SetTolerance(0.005);

vtkProperty* ipwProp = vtkProperty::New();
 //assign default props to the ipw's texture plane actor

vtkImagePlaneWidget* planeWidgetX = vtkImagePlaneWidget::New();
  planeWidgetX->SetInteractor( iren);
  planeWidgetX->SetKeyPressActivationValue('x');
  planeWidgetX->SetPicker(picker);
  planeWidgetX->RestrictPlaneToVolumeOn();
  planeWidgetX->GetPlaneProperty()->SetColor(1,0,0);
  planeWidgetX->SetTexturePlaneProperty(ipwProp);
  planeWidgetX->TextureInterpolateOff();
  planeWidgetX->SetResliceInterpolateToNearestNeighbour();
  planeWidgetX->SetInput(v16->GetOutput());
  planeWidgetX->SetPlaneOrientationToXAxes();
  //planeWidgetX->SetSliceIndex(32);
  planeWidgetX->DisplayTextOn();
  planeWidgetX->On();
  planeWidgetX->InteractionOff();
  planeWidgetX->InteractionOn();

vtkImagePlaneWidget* planeWidgetY = vtkImagePlaneWidget::New();
  planeWidgetY->SetInteractor( iren);
  planeWidgetY->SetKeyPressActivationValue('y');
  planeWidgetY->SetPicker(picker);
  planeWidgetY->GetPlaneProperty()->SetColor(1,1,0);
  planeWidgetY->SetTexturePlaneProperty(ipwProp);
  planeWidgetY->TextureInterpolateOn();
  planeWidgetY->SetResliceInterpolateToLinear();
  planeWidgetY->SetInput(v16->GetOutput());
  planeWidgetY->SetPlaneOrientationToYAxes();
  //planeWidgetY->SetSlicePosition(102.4);
  planeWidgetY->SetLookupTable( planeWidgetX->GetLookupTable());
  planeWidgetY->DisplayTextOn();
  planeWidgetY->UpdatePlacement();
  planeWidgetY->On();

vtkImagePlaneWidget* planeWidgetZ = vtkImagePlaneWidget::New();
  planeWidgetZ->SetInteractor( iren);
  planeWidgetZ->SetKeyPressActivationValue('z');
  planeWidgetZ->SetPicker(picker);
  planeWidgetZ->GetPlaneProperty()->SetColor(0,0,1);
  planeWidgetZ->SetTexturePlaneProperty(ipwProp);
  planeWidgetZ->TextureInterpolateOn();
  planeWidgetZ->SetResliceInterpolateToCubic();
  planeWidgetZ->SetInput(v16->GetOutput());
  planeWidgetZ->SetPlaneOrientationToZAxes();
  //planeWidgetZ->SetSliceIndex(25);
  planeWidgetZ->SetLookupTable( planeWidgetX->GetLookupTable());
  planeWidgetZ->DisplayTextOn();
  planeWidgetZ->On();

vtkImageOrthoPlanes *orthoPlanes = vtkImageOrthoPlanes::New();
  orthoPlanes->SetPlane(0, planeWidgetX);
  orthoPlanes->SetPlane(1, planeWidgetY);
  orthoPlanes->SetPlane(2, planeWidgetZ);
  orthoPlanes->ResetPlanes();

 vtkOrthoPlanesCallback* cbk = vtkOrthoPlanesCallback::New();
 cbk->WidgetX = planeWidgetX;
 cbk->WidgetY = planeWidgetY;
 cbk->WidgetZ = planeWidgetZ;
 planeWidgetX->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
 planeWidgetY->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
 planeWidgetZ->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
 cbk->Delete();
```

```
double wl[2];
planeWidgetZ->GetWindowLevel(wl);

// Add a 2D image to test the GetReslice method
//
vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
  colorMap->PassAlphaToOutputOff();
  colorMap->SetActiveComponent(0);
  colorMap->SetOutputFormatToLuminance();
  colorMap->SetInput(planeWidgetZ->GetResliceOutput());
  colorMap->SetLookupTable(planeWidgetX->GetLookupTable());

vtkImageActor* imageActor = vtkImageActor::New();
  imageActor->PickableOff();
  imageActor->SetInput(colorMap->GetOutput());

// Add the actors
//
ren1->AddActor( outlineActor);
ren2->AddActor( imageActor);

ren1->SetBackground( 0.1, 0.1, 0.2);
ren2->SetBackground( 0.2, 0.1, 0.2);

renWin->SetSize( 600, 350);

ren1->SetViewport(0,0,0.58333,1);
ren2->SetViewport(0.58333,0,1,1);

// Set the actors' postions
//
renWin->Render();
//iren->SetEventPosition( 175,175);
//iren->SetKeyCode('r');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//iren->SetEventPosition( 475,175);
//iren->SetKeyCode('r');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//renWin->Render();

//ren1->GetActiveCamera()->Elevation(110);
//ren1->GetActiveCamera()->SetViewUp(0, 0, -1);
//ren1->GetActiveCamera()->Azimuth(45);
//ren1->GetActiveCamera()->Dolly(1.15);
ren1->ResetCameraClippingRange();

vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText ( "R" );
cube->SetXMinusFaceText( "L" );
cube->SetYPlusFaceText ( "A" );
cube->SetYMinusFaceText( "P" );
cube->SetZPlusFaceText ( "H" );
cube->SetZMinusFaceText( "F" );
cube->SetFaceTextScale( 0.666667 );

vtkAxesActor* axes2 = vtkAxesActor::New();

vtkMatrix4x4 *invert = vtkMatrix4x4::New();
invert->DeepCopy( reader->GetDirectionCosines() );
invert->Invert();

// simulate a left-handed coordinate system
//
vtkTransform *transform = vtkTransform::New();
transform->Identity();
//transform->RotateY(90);
transform->Concatenate(invert);
axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
cube->GetAssembly()->SetUserTransform( transform );

axes2->SetTotalLength( 1.5, 1.5, 1.5 );
axes2->SetCylinderRadius( 0.500 * axes2->GetCylinderRadius() );
axes2->SetConeRadius    ( 1.025 * axes2->GetConeRadius() );
axes2->SetSphereRadius  ( 1.500 * axes2->GetSphereRadius() );

vtkTextProperty* tprop = axes2->GetXAxisCaptionActor2D()->
  GetCaptionTextProperty();
tprop->ItalicOn();
tprop->ShadowOn();
```

```
  tprop->SetFontFamilyToTimes();

  axes2->GetYAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop
       );
  axes2->GetZAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop
       );

  vtkPropAssembly* assembly = vtkPropAssembly::New();
  assembly->AddPart( axes2 );
  assembly->AddPart( cube );

  vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
  widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
  widget->SetOrientationMarker( assembly );
  widget->SetInteractor( iren );
  widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
  widget->SetEnabled( 1 );
  widget->InteractiveOff();
  widget->InteractiveOn();

  // Playback recorded events
  //
  //vtkInteractorEventRecorder *recorder = vtkInteractorEventRecorder::New();
  //recorder->SetInteractor(iren);
  //recorder->ReadFromInputStringOn();
  //recorder->SetInputString(IOPeventLog);

  // Interact with data
  // Render the image
  //
  iren->Initialize();
  renWin->Render();

  // Test SetKeyPressActivationValue for one of the widgets
  //
  //iren->SetKeyCode('z');
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);
  //iren->SetKeyCode('z');
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);

  //int retVal = vtkRegressionTestImage( renWin );
  //
  //if ( retVal == vtkRegressionTester::DO_INTERACTOR)
    {
    iren->Start();
    }

  // Clean up
  //
  //recorder->Off();
  //recorder->Delete();

  ipwProp->Delete();
  orthoPlanes->Delete();
  planeWidgetX->Delete();
  planeWidgetY->Delete();
  planeWidgetZ->Delete();
  colorMap->Delete();
  imageActor->Delete();
  picker->Delete();
  outlineActor->Delete();
  outlineMapper->Delete();
  outline->Delete();
  iren->Delete();
  renWin->Delete();
  ren1->Delete();
  ren2->Delete();
  v16->Delete();
  reader->Delete();

  return 0;
}
```

27.48 gdcmreslice.cxx

```
/*=========================================================================
```

```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkImageFlip.h"
#include "vtkImageReslice.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"

int main( int argc, char *argv[] )
{
  if( argc < 2 ) return 1;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New
      ();
  reader->SetFileName( argv[1] );
  //reader->FileLowerLeftOn();
  reader->Update();

  vtkImageFlip *flip = vtkImageFlip::New();
  flip->SetInput(reader->GetOutput());
  flip->SetFilteredAxis(0);
  flip->Update();

  vtkImageReslice *reslice = vtkImageReslice::New();
  //reslice->SetInput(reader->GetOutput());
  reslice->SetInput(flip->GetOutput());
  //reslice->SetResliceAxesDirectionCosines()
  reader->GetDirectionCosines()->Print(std::cout);
  vtkMatrix4x4 *invert = vtkMatrix4x4::New();
  invert->DeepCopy( reader->GetDirectionCosines() );
  invert->Invert();

  //reslice->SetResliceAxes( reader->GetDirectionCosines() );
  reslice->SetResliceAxes( invert );
  reslice->Update();
  vtkImageData* ima = reslice->GetOutput();

  vtkLookupTable* table = vtkLookupTable::New();
  table->SetNumberOfColors(1000);
  table->SetTableRange(0,1000);
  table->SetSaturationRange(0,0);
  table->SetHueRange(0,1);
  table->SetValueRange(0,1);
  table->SetAlphaRange(1,1);
  table->Build();

  // Texture
  vtkTexture* texture = vtkTexture::New();
  texture->SetInput(ima);
  texture->InterpolateOn();
  texture->SetLookupTable(table);

  // PlaneSource
  vtkPlaneSource* plane = vtkPlaneSource::New();

  // PolyDataMapper
  vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
```

```
  planeMapper->SetInput(plane->GetOutput());

  // Actor
  vtkActor* planeActor = vtkActor::New();
  planeActor->SetTexture(texture);
  planeActor->SetMapper(planeMapper);
  planeActor->PickableOn();

  // Final rendering with simple interactor:
  vtkRenderer        *ren = vtkRenderer::New();
  vtkRenderWindow *renwin = vtkRenderWindow::New();
  renwin->AddRenderer(ren);
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renwin);
  ren->AddActor(planeActor);
  ren->SetBackground(0,0,0.5);

  // DICOM is RAH:
  vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
  cube->SetXPlusFaceText ( "R" );
  cube->SetXMinusFaceText( "L" );
  cube->SetYPlusFaceText ( "A" );
  cube->SetYMinusFaceText( "P" );
  cube->SetZPlusFaceText ( "H" );
  cube->SetZMinusFaceText( "F" );

  vtkAxesActor* axes2 = vtkAxesActor::New();

  vtkTransform *transform = vtkTransform::New();
  transform->Identity();
  //reader->GetDirectionCosines()->Print(std::cout);
  transform->Concatenate(invert);
  //axes2->SetShaftTypeToCylinder();
  axes2->SetUserTransform( transform );
  cube->GetAssembly()->SetUserTransform( transform ); // cant get it to work

  vtkPropAssembly* assembly = vtkPropAssembly::New();
  assembly->AddPart( axes2 );
  assembly->AddPart( cube );

  vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
  widget->SetOrientationMarker( assembly );
  widget->SetInteractor( iren );
  widget->SetEnabled( 1 );
  widget->InteractiveOff();
  widget->InteractiveOn();

  renwin->Render();
  iren->Start();

  // Clean up:
  reader->Delete();
  table->Delete();
  texture->Delete();
  plane->Delete();
  planeMapper->Delete();
  planeActor->Delete();
  ren->Delete();
  renwin->Delete();
  iren->Delete();

  return 0;
}
```

27.49 gdcmrtionplan.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.
```

```
=========================================================================*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkXMLPolyDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>

#include "gdcmReader.h"
#include "gdcmAttribute.h"

/*
 This example is just for fun. We found a RT Ion Plan Storage and simply
       extracted the viz stuff for VTK

    RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " filename.dcm outfile.vti\n";
    return 1;
    }
  const char * filename = argv[1];
  const char * outfilename = argv[2];
  const char * outfilename2 = argv[3];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::MediaStorage ms;
  ms.SetFromFile( reader.GetFile() );
  if( ms != gdcm::MediaStorage::RTIonPlanStorage
      )
    {
    return 1;
    }

/*
(300a,03a2) SQ                                                  # u/l,1 Ion
       Beam Sequence
  (fffe,e000) na (Item with undefined length)
    (0008,1040) LO [Test]                                       # 4,1
       Institutional Department Name
    (300a,00b2) SH (no value)                                   # 0,1
       Treatment Machine Name
    (300a,00b3) CS [MU]                                         # 2,1 Primary
       Dosimeter Unit
    (300a,00c0) IS [1 ]                                         # 2,1 Beam
       Number
    (300a,00c2) LO [1 ]                                         # 2,1 Beam
       Name
    (300a,00c4) CS [STATIC]                                     # 6,1 Beam
       Type
    (300a,00c6) CS [PROTON]                                     # 6,1
       Radiation Type
    (300a,00ce) CS [TREATMENT ]                                 # 10,1
       Treatment Delivery Type
    (300a,00d0) IS [0 ]                                         # 2,1 Number
       of Wedges
    (300a,00e0) IS [1 ]                                         # 2,1 Number
       of Compensators
    (300a,00ed) IS [0 ]                                         # 2,1 Number
       of Boli
    (300a,00f0) IS [1 ]                                         # 2,1 Number
       of Blocks
    (300a,0110) IS [2 ]                                         # 2,1 Number
```

```
      of Control Points
   (300a,02ea) SQ                                          # u/l,1 Ion
      Range Compensator Sequence
     (fffe,e000) na (Item with undefined length)
       (300a,00e1) SH [lucite]                             # 6,1
      Material ID
       (300a,00e4) IS [1 ]                                 # 2,1
      Compensator Number
       (300a,00e5) SH [75hdhe5 ]                           # 8,1
      Compensator ID
       (300a,00e7) IS [35]                                 # 2,1
      Compensator Rows
       (300a,00e8) IS [37]                                 # 2,1
      Compensator Columns
       (300a,00e9) DS [3.679991\4.249288 ]                 # 18,2
      Compensator Pixel Spacing
       (300a,00ea) DS [-76.00\62.50]                       # 12,2
      Compensator Position
       (300a,00ec) DS
      [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38
     .79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.10\38.62\44.8
     8\44.88\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77\39.77\43.52\52.13\52.13\52.13\53.18\53.52\54.0]
      Thickness Data
       (300a,02e0) CS [ABSENT]                             # 6,1
      Compensator Divergence
       (300a,02e1) CS [SOURCE_SIDE ]                       # 12,1
      Compensator Mounting Position
       (300a,02e4) FL 39.2                                 # 4,1
      Isocenter to Compensator Tray Distance
       (300a,02e5) FL 2.12                                 # 4,1
      Compensator Column Offset
       (300a,02e8) FL 4.76                                 # 4,1
      Compensator Milling Tool Diameter
     (fffe,e00d)
*/
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet
      ();
  gdcm::Tag tbeamsq(0x300a,0x03a2);
  if( !ds.FindDataElement( tbeamsq ) )
    {
    return 1;
    }
  const gdcm::DataElement &beamsq = ds.GetDataElement
      ( tbeamsq );
  //std::cout << beamsq << std::endl;
  gdcm::SmartPointer<gdcm::SequenceOfItems>
       sqi = beamsq.GetValueAsSQ();
  if( !sqi || !sqi->GetNumberOfItems() )
    {
    return 1;
    }

  //for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
  //  {
    //const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
    const gdcm::Item & item = sqi->GetItem(1); // Item start
       at #1
    const gdcm::DataSet& nestedds = item.GetNestedDataSet
      ();
    //std::cout << nestedds << std::endl;
    gdcm::Tag tcompensatorsq(0x300a,0x02ea);
    if( !nestedds.FindDataElement( tcompensatorsq ) )
      {
      return 1;
      }
    const gdcm::DataElement &compensatorsq = nestedds.
      GetDataElement( tcompensatorsq );
    //std::cout << compensatorsq << std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems>
       ssqi = compensatorsq.GetValueAsSQ();
    const gdcm::Item & item2 = ssqi->GetItem(1); // Item start
       at #1
    const gdcm::DataSet& nestedds2 = item2.GetNestedDataSet
      ();
    //std::cout << nestedds2 << std::endl;
    gdcm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
    if( !nestedds2.FindDataElement( tcompensatorthicknessdata )
      )
      {
      return 1;
      }
```

```
    const gdcm::DataElement &compensatorthicknessdata =
      nestedds2.GetDataElement( tcompensatorthicknessdata );
    //  std::cout << compensatorthicknessdata << std::endl;
    gdcm::Attribute<0x300a,0x00ec> at;
    at.SetFromDataElement( compensatorthicknessdata );
    const double* pts = at.GetValues();
    //        (300a,00e7) IS [35]                                     # 2,1
       Compensator Rows
    gdcm::Attribute<0x300a,0x00e7> at1;
    const gdcm::DataElement &compensatorrows = nestedds2.
      GetDataElement( at1.GetTag() );
    at1.SetFromDataElement( compensatorrows );
    std::cout << at1.GetValue() << std::endl;
    //        (300a,00e8) IS [37]                                     # 2,1
       Compensator Columns
    gdcm::Attribute<0x300a,0x00e8> at2;
    const gdcm::DataElement &compensatorcols = nestedds2.
      GetDataElement( at2.GetTag() );
    at2.SetFromDataElement( compensatorcols );
    std::cout << at2.GetValue() << std::endl;

        // (300a,00e9) DS [3.679991\4.249288 ]                      # 18,2
       Compensator Pixel Spacing
    gdcm::Attribute<0x300a,0x00e9> at3;
    const gdcm::DataElement &compensatorpixelspacing =
      nestedds2.GetDataElement( at3.GetTag() );
    at3.SetFromDataElement( compensatorpixelspacing );
    std::cout << at3.GetValue(0) << std::endl;
        // (300a,00ea) DS [-76.00\62.50]                            # 12,2
       Compensator Position
    gdcm::Attribute<0x300a,0x00ea> at4;
    const gdcm::DataElement &compensatorposition = nestedds2.
      GetDataElement( at4.GetTag() );
    at4.SetFromDataElement( compensatorposition );
    std::cout << at4.GetValue(0) << std::endl;

    vtkDoubleArray *d = vtkDoubleArray::New();
    d->SetArray( (double*)pts , at1.GetValue() * at2.GetValue()
       , 0 );

    vtkImageData *img = vtkImageData::New();
    img->Initialize();
    img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
    //imgb->SetExtent(1, xdim, 1, ydim, 1, zdim);
    img->SetScalarTypeToDouble();
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); //
       FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
    img->SetNumberOfScalarComponents(1);
    img->GetPointData()->SetScalars(d);

    img->Update();
    img->Print(std::cout);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
    writeb->SetInput( img );
    writeb->SetFileName( outfilename );
    writeb->Write( );
/*
    (300a,03a6) SQ                                      # u/1,1 Ion Block
       Sequence
      (fffe,e000) na (Item with undefined length)
        (300a,00e1) SH [brass ]                         # 6,1 Material ID
        (300a,00f7) FL 95.03                            # 4,1 Isocenter to
       Block Tray Distance
        (300a,00f8) CS [APERTURE]                       # 8,1 Block Type
        (300a,00fa) CS [ABSENT]                         # 6,1 Block
       Divergence
        (300a,00fb) CS [SOURCE_SIDE ]                   # 12,1 Block Mounting
       Position
        (300a,00fc) IS [1 ]                             # 2,1 Block Number
        (300a,0100) DS [50.00 ]                         # 6,1 Block Thickness
        (300a,0104) IS [179 ]                           # 4,1 Block Number of
       Points
        (300a,0106) DS
       [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47
      .1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2\44.6\28.9\44
      .2\29.7\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7\40.0\39.5\41.5\37.9\42
2\37.4\43.0\37.1\44.7\36] # 1934,2-2n Block Data
      (fffe,e00d)
    (fffe,e0dd)
```

```
*/
    gdcm::Tag tblocksq(0x300a,0x03a6);
    if( !nestedds.FindDataElement( tblocksq ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocksq = nestedds.GetDataElement
      ( tblocksq );
    //std::cout << blocksq << std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems>
       sssqi = blocksq.GetValueAsSQ();
    const gdcm::Item & item3 = sssqi->GetItem(1); // Item
       start at #1
    const gdcm::DataSet& nestedds3 = item3.GetNestedDataSet
      ();

    gdcm::Tag tblockdata(0x300a,0x0106);
    if( !nestedds3.FindDataElement( tblockdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &blockdata = nestedds3.
      GetDataElement( tblockdata );
    //  std::cout << blockdata << std::endl;
    gdcm::Attribute<0x300a,0x0106> at_;
    at_.SetFromDataElement( blockdata );

    vtkDoubleArray *scalars = vtkDoubleArray::New();
    scalars->SetNumberOfComponents(3);

    gdcm::Attribute<0x300a,0x0104> bnpts; //  IS
       [179 ]                                    # 4,1 Block Number of Points
    if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocknpts = nestedds3.
      GetDataElement( bnpts.GetTag() );
    bnpts.SetFromDataElement(  blocknpts );
    //std::cout << bnpts.GetValue() << std::endl;

    vtkPolyData *output = vtkPolyData::New();
    vtkPoints *newPts = vtkPoints::New();
    vtkCellArray *polys = vtkCellArray::New();
    const double *ptr = at_.GetValues();
    //unsigned int npts = bnpts.GetNumberOfValues() / 2;
    unsigned int npts = bnpts.GetValue();
    vtkIdType *ptIds =  new vtkIdType[npts];
    for(unsigned int i = 0; i < npts; ++i)
      {
      float x[3] = {};
      x[0] = ptr[2*i+0];
      x[1] = ptr[2*i+1];
      //x[2] = pts[i+2];
      vtkIdType ptId = newPts->InsertNextPoint( x );
      //std::cout << x[0] << "," << x[1] << "," << x[2] << std::endl;
      ptIds[i ] = ptId;
      }
    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    (void)cellId;
    delete[] ptIds;

    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
    output->Update();
    output->Print( std::cout );



  //  }

   vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

   vtkImageColorViewer *viewer = vtkImageColorViewer::New
      ();
```

```
   viewer->SetInput(img);
   viewer->SetupInteractor(iren);
   viewer->SetSize(600, 600);
   viewer->GetRenderer()->ResetCameraClippingRange();
   viewer->Render();
   viewer->GetRenderer()->ResetCameraClippingRange();

  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
      cubeMapper->SetInput( output );
      cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  //vtkActor2D* cubeActor = vtkActor2D::New();
      cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();

viewer->GetRenderer()->AddActor( cubeActor );

    vtkXMLPolyDataWriter *writec= vtkXMLPolyDataWriter::New();
    writec->SetInput( output );
    writec->SetFileName( outfilename2 );
    writec->Write( );

   iren->Initialize();
   iren->Start();


  return 0;
}
```

27.50 gdcmrtplan.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>

#include "gdcmReader.h"
#include "gdcmAttribute.h"

/*
 This example is just for fun. We found a fake RT Ion Plan Storage and simply
       extracted the viz stuff for VTK
 but this is rather a RT Plan storage
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " filename.dcm outfile.vti\n";
    return 1;
    }
  const char * filename = argv[1];
  const char * outfilename = argv[2];
```

```
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::MediaStorage ms;
  ms.SetFromFile( reader.GetFile() );
  if( ms != gdcm::MediaStorage::RTIonPlanStorage
      )
    {
    return 1;
    }

/*
(300a,00b0) SQ                                                  # u/l,1 Beam
       Sequence
  (fffe,e000) na (Item with undefined length)
    (300a,00b2) SH (no value)                                   # 0,1
       Treatment Machine Name
    (300a,00c0) IS [1 ]                                         # 2,1 Beam
       Number
    (300a,00c2) LO [1 ]                                         # 2,1 Beam
       Name
    (300a,00c4) CS [STATIC]                                     # 6,1 Beam
       Type
    (300a,00c6) CS [PROTON]                                     # 6,1
       Radiation Type
    (300a,00ce) CS [TREATMENT ]                                 # 10,1
       Treatment Delivery Type
    (300a,00e0) IS [1 ]                                         # 2,1 Number
       of Compensators
    (300a,00e3) SQ                                              # u/l,1
       Compensator Sequence
      (fffe,e000) na (Item with undefined length)
        (300a,00e1) SH [lucite]                                 # 6,1
       Material ID
        (300a,00e4) IS [1 ]                                     # 2,1
       Compensator Number
        (300a,00e5) SH [75hdhe5 ]                               # 8,1
       Compensator ID
        (300a,00e7) IS [35]                                     # 2,1
       Compensator Rows
        (300a,00e8) IS [37]                                     # 2,1
       Compensator Columns
        (300a,00e9) DS [3.679991\4.249288 ]                     # 18,2
       Compensator Pixel Spacing
        (300a,00ea) DS [-76.00\62.50]                           # 12,2
       Compensator Position
        (300a,00ec) DS
       [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38
      .79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.10\38.62\44.8
      8\44.88\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77\39.77\43.52\52.13\52.13\52.13\53.18\53.52\54.0]
       Thickness Data
        (300a,02e0) CS [ABSENT]                                 # 6,1
       Compensator Divergence
        (300a,02e1) CS [SOURCE_SIDE ]                           # 12,1
       Compensator Mounting Position
      (fffe,e00d)
      (fffe,e000) na (Item with undefined length)
      (fffe,e00d)
    (fffe,e0dd)
*/
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet
      ();
  gdcm::Tag tbeamsq(0x300a,0x00b0);
  if( !ds.FindDataElement( tbeamsq ) )
    {
    return 1;
    }
  const gdcm::DataElement &beamsq = ds.GetDataElement
      ( tbeamsq );
  //std::cout << beamsq << std::endl;
  gdcm::SmartPointer<gdcm::SequenceOfItems>
      sqi = beamsq.GetValueAsSQ();
  if( !sqi || !sqi->GetNumberOfItems() )
    {
    return 1;
    }
```

```
//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
//  {
  //const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
  const gdcm::Item & item = sqi->GetItem(2); // Item start
     at #1
  const gdcm::DataSet& nestedds = item.GetNestedDataSet
    ();
  //std::cout << nestedds << std::endl;
  gdcm::Tag tcompensatorsq(0x300a,0x00e3);
  if( !nestedds.FindDataElement( tcompensatorsq ) )
    {
    return 1;
    }
  const gdcm::DataElement &compensatorsq = nestedds.
    GetDataElement( tcompensatorsq );
  //std::cout << compensatorsq << std::endl;
  gdcm::SmartPointer<gdcm::SequenceOfItems>
     ssqi = compensatorsq.GetValueAsSQ();
  const gdcm::Item & item2 = ssqi->GetItem(1); // Item start
     at #1
  const gdcm::DataSet& nestedds2 = item2.GetNestedDataSet
    ();
  //std::cout << nestedds2 << std::endl;
  gdcm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
  if( !nestedds2.FindDataElement( tcompensatorthicknessdata )
    )
    {
    return 1;
    }
  const gdcm::DataElement &compensatorthicknessdata =
    nestedds2.GetDataElement( tcompensatorthicknessdata );
  //  std::cout << compensatorthicknessdata << std::endl;
  gdcm::Attribute<0x300a,0x00ec> at;
  at.SetFromDataElement( compensatorthicknessdata );
  const double* pts = at.GetValues();
  //        (300a,00e7) IS [35]                                    # 2,1
     Compensator Rows
  gdcm::Attribute<0x300a,0x00e7> at1;
  const gdcm::DataElement &compensatorrows = nestedds2.
    GetDataElement( at1.GetTag() );
  at1.SetFromDataElement( compensatorrows );
  std::cout << at1.GetValue() << std::endl;
  //        (300a,00e8) IS [37]                                    # 2,1
     Compensator Columns
  gdcm::Attribute<0x300a,0x00e8> at2;
  const gdcm::DataElement &compensatorcols = nestedds2.
    GetDataElement( at2.GetTag() );
  at2.SetFromDataElement( compensatorcols );
  std::cout << at2.GetValue() << std::endl;

      // (300a,00e9) DS [3.679991\4.249288 ]                      # 18,2
     Compensator Pixel Spacing
  gdcm::Attribute<0x300a,0x00e9> at3;
  const gdcm::DataElement &compensatorpixelspacing =
    nestedds2.GetDataElement( at3.GetTag() );
  at3.SetFromDataElement( compensatorpixelspacing );
  std::cout << at3.GetValue(0) << std::endl;
      // (300a,00ea) DS [-76.00\62.50]                            # 12,2
     Compensator Position
  gdcm::Attribute<0x300a,0x00ea> at4;
  const gdcm::DataElement &compensatorposition = nestedds2.
    GetDataElement( at4.GetTag() );
  at4.SetFromDataElement( compensatorposition );
  std::cout << at4.GetValue(0) << std::endl;

  vtkDoubleArray *d = vtkDoubleArray::New();
  d->SetArray( (double*)pts , at1.GetValue() * at2.GetValue()
     , 0 );

  vtkImageData *img = vtkImageData::New();
  img->Initialize();
  img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
  //imgb->SetExtent(1, xdim, 1, ydim, 1, zdim);
  img->SetScalarTypeToDouble();
  img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); //
     FIXME image is upside down
  img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
  img->SetNumberOfScalarComponents(1);
  img->GetPointData()->SetScalars(d);
```

```
    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
    writeb->SetInput( img );
    writeb->SetFileName( outfilename );
    writeb->Write( );
/*
 (300a,00f4) SQ                                          # u/l,1 Block
       Sequence
      (fffe,e000) na (Item with undefined length)
        (300a,00e1) SH [brass ]                              # 6,1
       Material ID
        (300a,00f8) CS [APERTURE]                            # 8,1 Block
       Type
        (300a,00fa) CS [ABSENT]                              # 6,1 Block
       Divergence
        (300a,00fb) CS [SOURCE_SIDE ]                        # 12,1 Block
       Mounting Position
        (300a,00fc) IS [1 ]                                  # 2,1 Block
       Number
        (300a,0100) DS [50.00 ]                              # 6,1 Block
       Thickness
        (300a,0104) IS [179 ]                                # 4,1 Block
       Number of Points
        (300a,0106) DS
       [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47
      .1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2\44.6\28.9\44
      .2\29.7\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7\40.0\39.5\41.5\37.9\42.2\37.4\43.0\37.1\44.7\36]
      (fffe,e00d)
      (fffe,e000) na (Item with undefined length)
      (fffe,e00d)
    (fffe,e0dd)
*/
    gdcm::Tag tblocksq(0x300a,0x00f4);
    if( !nestedds.FindDataElement( tblocksq ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocksq = nestedds.GetDataElement
      ( tblocksq );
    //std::cout << blocksq << std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems>
       sssqi = blocksq.GetValueAsSQ();
    const gdcm::Item & item3 = sssqi->GetItem(1); // Item
       start at #1
    const gdcm::DataSet& nestedds3 = item3.GetNestedDataSet
      ();

    gdcm::Tag tblockdata(0x300a,0x0106);
    if( !nestedds3.FindDataElement( tblockdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &blockdata = nestedds3.
      GetDataElement( tblockdata );
    //  std::cout << blockdata << std::endl;
    gdcm::Attribute<0x300a,0x0106> at_;
    at_.SetFromDataElement( blockdata );

    vtkDoubleArray *scalars = vtkDoubleArray::New();
    scalars->SetNumberOfComponents(3);

    gdcm::Attribute<0x300a,0x0104> bnpts; //  IS
       [179 ]                                     # 4,1 Block Number of Points
    if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocknpts = nestedds3.
      GetDataElement( bnpts.GetTag() );
    bnpts.SetFromDataElement(  blocknpts );
  std::cout << bnpts.GetValue() << std::endl;

    vtkPolyData *output = vtkPolyData::New();
    vtkPoints *newPts = vtkPoints::New();
    vtkCellArray *polys = vtkCellArray::New();
    const double *ptr = at_.GetValues();
      //unsigned int npts = bnpts.GetNumberOfValues() / 2;
      unsigned int npts = bnpts.GetValue();
    vtkIdType *ptIds =  new vtkIdType[npts];
    for(unsigned int i = 0; i < npts; ++i)
      {
        float x[3] = {};
```

```
        x[0] = ptr[2*i+0];
        x[1] = ptr[2*i+1];
        //x[2] = pts[i+2];
      vtkIdType ptId = newPts->InsertNextPoint( x );
      //std::cout << x[0] << "," << x[1] << "," << x[2] << std::endl;
        ptIds[i ] = ptId;
      }
      vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
        delete[] ptIds;

    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
    output->Update();
    output->Print( std::cout );




  //  }

   vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

   vtkImageColorViewer *viewer = vtkImageColorViewer::New
       ();
   viewer->SetInput(img);
   viewer->SetupInteractor(iren);
   viewer->SetSize(600, 600);
   viewer->Render();

  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
      cubeMapper->SetInput( output );
      cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  //vtkActor2D* cubeActor = vtkActor2D::New();
      cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();

  viewer->GetRenderer()->AddActor( cubeActor );

   iren->Initialize();
   iren->Start();


  return 0;
}
```

27.51 gdcmscene.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMPolyDataReader.h"
//#include "vtkGDCMPolyDataWriter.h"

#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderWindow.h"
```

```
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"


// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr << argv[0] << " filename1.dcm\n";
    return 1;
    }
  const char * filename = argv[1];

  vtkGDCMPolyDataReader * reader =
      vtkGDCMPolyDataReader::New();
  reader->SetFileName( filename );
  reader->Update();

//  vtkGDCMPolyDataWriter * writer2 = vtkGDCMPolyDataWriter::New();
//  for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
//    writer2->SetInput( num, reader->GetOutput(num) );
//  writer2->SetFileName( "rtstruct.dcm" );
//  writer2->Write();

  // print reader output:
  reader->Print( std::cout );
  // print first output:
  reader->GetOutput()->Print( std::cout );

  vtkAppendPolyData *append = vtkAppendPolyData::New();
  int n = reader->GetNumberOfOutputPorts();
  for(int i = 0; i < n; ++i)
    {
    append->AddInput( reader->GetOutput(i) );
    }

  vtkPolyDataWriter * writer = vtkPolyDataWriter::New();
  writer->SetInput( reader->GetOutput() );
  writer->SetFileName( "rtstruct.vtk" );
  //writer->Write();

  // Now we'll look at it.
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
      //cubeMapper->SetInput( reader->GetOutput() );
      cubeMapper->SetInput( append->GetOutput() );
      cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  //vtkActor2D* cubeActor = vtkActor2D::New();
      cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();
  //cubeActor->GetProperty()->SetColor(1, 0, 0);


  // The usual rendering stuff.
//  vtkCamera *camera = vtkCamera::New();
//      camera->SetPosition(1,1,1);
//      camera->SetFocalPoint(0,0,0);

  vtkRenderer *renderer = vtkRenderer::New();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
    renWin->AddRenderer(renderer);

  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

  renderer->AddActor(cubeActor);
  //renderer->AddActor2D(cubeActor);
      //renderer->SetActiveCamera(camera);
      renderer->ResetCamera();
      renderer->SetBackground(1,1,1);

  renWin->SetSize(300,300);
```

```
  // interact with data
  renWin->Render();
  iren->Start();



  reader->Delete();
  append->Delete();
  cubeMapper->Delete();
  cubeActor->Delete();
//  camera->Delete();
  renderer->Delete();
  renWin->Delete();
  iren->Delete();

  writer->Delete();

  return 0;
}
```

27.52 gdcmtexture.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"

int main( int argc, char *argv[] )
{
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New
      ();
  reader->SetFileName( argv[1] );

  reader->Update();
  vtkImageData* ima = reader->GetOutput();

  vtkLookupTable* table = vtkLookupTable::New();
  table->SetNumberOfColors(1000);
  table->SetTableRange(0,1000);
  table->SetSaturationRange(0,0);
  table->SetHueRange(0,1);
  table->SetValueRange(0,1);
  table->SetAlphaRange(1,1);
  table->Build();

  // Texture
  vtkTexture* texture = vtkTexture::New();
  texture->SetInput(ima);
  texture->InterpolateOn();
```

```
  texture->SetLookupTable(table);

  // PlaneSource
  vtkPlaneSource* plane = vtkPlaneSource::New();
  plane->SetOrigin( -0.5, -0.5, 0.0);
  plane->SetPoint1(  0.5, -0.5, 0.0);
  plane->SetPoint2( -0.5,  0.5, 0.0);

  // PolyDataMapper
  vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
  planeMapper->SetInput(plane->GetOutput());

  // Actor
  vtkActor* planeActor = vtkActor::New();
  planeActor->SetTexture(texture);
  planeActor->SetMapper(planeMapper);
  planeActor->PickableOn();

  // Final rendering with simple interactor:
  vtkRenderer        *ren = vtkRenderer::New();
  vtkRenderWindow *renwin = vtkRenderWindow::New();
  renwin->AddRenderer(ren);
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renwin);
  ren->AddActor(planeActor);
  ren->SetBackground(0,0,0.5);

  vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
  cube->SetXPlusFaceText ( "L" );
  cube->SetXMinusFaceText( "R" );
  cube->SetYPlusFaceText ( "A" );
  cube->SetYMinusFaceText( "P" );
  cube->SetZPlusFaceText ( "H" );
  cube->SetZMinusFaceText( "F" );

  vtkAxesActor* axes2 = vtkAxesActor::New();
  // simulate a left-handed coordinate system
  //
  vtkTransform *transform = vtkTransform::New();
  transform->Identity();
  //transform->RotateY(180);
  reader->GetDirectionCosines()->Print(std::cout);
  transform->Concatenate(reader->GetDirectionCosines());
  //axes2->SetShaftTypeToCylinder();
  axes2->SetUserTransform( transform );
  //cube->SetUserTransform( transform ); // cant get it to work
  cube->GetAssembly()->SetUserTransform( transform ); // cant get it to work

  vtkPropAssembly* assembly = vtkPropAssembly::New();
  assembly->AddPart( axes2 );
  assembly->AddPart( cube );

  vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
  //widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
  widget->SetOrientationMarker( assembly );
  widget->SetInteractor( iren );
  //widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
  widget->SetEnabled( 1 );
  widget->InteractiveOff();
  widget->InteractiveOn();

  renwin->Render();
  iren->Start();

  // Clean up:
  reader->Delete();
  table->Delete();
  texture->Delete();
  plane->Delete();
  planeMapper->Delete();
  planeActor->Delete();
  ren->Delete();
  renwin->Delete();
  iren->Delete();

  return 0;
}
```

27.53 gdcmvolume.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolume.h"
#include "vtkVolumeProperty.h"
#include "vtkVolumeTextureMapper3D.h"
#include "vtkFixedPointVolumeRayCastMapper.h"
#include "vtkInteractorStyleTrackballCamera.h"
#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkImageClip.h"
#include "vtkRenderWindowInteractor.h"


// gdcmvolume gdcmData/GE_DLX-8-MONO2-Multiframe-Jpeg_Lossless.dcm
int main(int argc, char *argv[])
{
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New
      ();
  reader->SetFileName( argv[1] );
  reader->Update();

  // Create the renderers, render window, and interactor
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renWin);
  vtkRenderer *ren = vtkRenderer::New();
  renWin->AddRenderer(ren);

  // Create a transfer function mapping scalar value to opacity
  vtkPiecewiseFunction *oTFun = vtkPiecewiseFunction::New();
  //oTFun->AddSegment(0, 1.0, 256, 0.1);
  oTFun->AddSegment(0, 1.0, 240, 0.1);

  vtkColorTransferFunction *cTFun = vtkColorTransferFunction::New();
  cTFun->AddRGBPoint(   0, 1.0, 1.0, 1.0 );
  //cTFun->AddRGBPoint( 255, 1.0, 1.0, 1.0 );
  cTFun->AddRGBPoint( 240, 1.0, 1.0, 1.0 );

  // Need to crop to actually see minimum intensity
  vtkImageClip *clip = vtkImageClip::New();
  clip->SetInputConnection( reader->GetOutputPort() );
  clip->SetOutputWholeExtent(0,66,0,66,30,37);
  clip->ClipDataOn();

  vtkVolumeProperty *property = vtkVolumeProperty::New();
  property->SetScalarOpacity(oTFun);
  property->SetColor(cTFun);
  property->SetInterpolationTypeToLinear();

  vtkFixedPointVolumeRayCastMapper *mapper =
      vtkFixedPointVolumeRayCastMapper::New();
  mapper->SetBlendModeToMinimumIntensity();
  mapper->SetInputConnection( reader->GetOutputPort() );

  vtkVolume *volume = vtkVolume::New();
  volume->SetMapper(mapper);
  volume->SetProperty(property);


  ren->AddViewProp(volume);

  renWin->Render();
    {
    iren->Start();
```

```
    }

  volume->Delete();
  mapper->Delete();
  property->Delete();
  clip->Delete();
  cTFun->Delete();
  oTFun->Delete();
  reader->Delete();
  renWin->Delete();
  iren->Delete();
  ren->Delete();


 return 0;
}
```

27.54 GenAllVR.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmDicts.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFileExplicitFilter.h"

#include <cstdlib>
#include <cstring>


gdcm::Tag FindTagFromVR(gdcm::Dict const &dict, gdcm::VR
       const &vr)
{
  using gdcm::Dict;
  Dict::ConstIterator beg = dict.Begin();
  Dict::ConstIterator end = dict.End();
  Dict::ConstIterator it;
  for( it = beg; it != end; ++it)
    {
    const gdcm::Tag &t = it->first;
    const gdcm::DictEntry &de = it->second;
    const gdcm::VR &vr_de = de.GetVR();
    if( vr == vr_de && !de.GetRetired() && t.GetGroup() >=
      0x8 )
      {
      return t;
      }
    }
  return gdcm::Tag(0xffff,0xffff);
}

struct rnd_gen {
  rnd_gen(char const* range = "abcdefghijklmnopqrstuvwxyz0123456789")
    : range(range), len(std::strlen(range)) { }
```

```
  char operator ()() const {
    return range[static_cast<std::size_t>(std::rand() * (1.0 / (RAND_MAX + 1.0
      )) * len)];
  }
private:
  char const* range;
  std::size_t len;
};

/*
 */
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr << argv[0] << " output.dcm" << std::endl;
    return 1;
    }
  const char *outfilename = argv[1];
  static const gdcm::Global &g = gdcm::Global::GetInstance
      ();
  static const gdcm::Dicts &dicts = g.GetDicts();
  static const gdcm::Dict &pubdict = dicts.GetPublicDict
      ();
  using gdcm::VR;
  using gdcm::Tag;

  gdcm::Writer w;

  gdcm::File &f = w.GetFile();
  gdcm::DataSet &ds = f.GetDataSet();

  gdcm::FileExplicitFilter fef;
  //fef.SetChangePrivateTags( true );
  fef.SetFile( w.GetFile() );
  if( !fef.Change() )
    {
    std::cerr << "Failed to change" << std::endl;
    return 1;
    }

  gdcm::SmartPointer<gdcm::SequenceOfItems>
       sq = new gdcm::SequenceOfItems();
  sq->SetLengthToUndefined();

//  gdcm::DummyValueGenerator dvg;

  const std::size_t len = 10;
  char ss[len+1];
  ss[len] = '\0';

  const char owner_str[] = "GDCM CONFORMANCE TESTS";
  gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
  owner.SetByteValue(owner_str, strlen(owner_str));
  owner.SetVR( gdcm::VR::LO );

  // Create an item
  gdcm::Item it;
  it.SetVLToUndefined();
  gdcm::DataSet &nds = it.GetNestedDataSet();
  //    nds.Insert(owner);
  //    nds.Insert(de);

  // Insert sequence into data set
  gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();

  ds.Insert(owner);
  ds.Insert(des);

  // avoid INVALID = 0
  for(int i = 1; i < 27; ++i)
    {
    VR vr = (VR::VRType)(1 << i);
    Tag t = FindTagFromVR( pubdict, vr );
    if( vr != VR::UN && vr != VR::SQ )
      {
      assert( t != Tag(0xffff,0xffff) );
```

```
      gdcm::DataElement de( t );
      std::generate_n(ss, len, rnd_gen());
      de.SetVR( vr );
      de.SetByteValue( ss, std::strlen( ss ) );
      nds.Insert( de );
      }
    }
  sq->AddItem(it);

  // Make sure to override any UID stuff
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );

  de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage
       );
  de.SetByteValue( ms.GetString(),strlen(ms.GetString()));
  ds.Insert( de );

  gdcm::FileMetaInformation &fmi = f.GetHeader
      ();
  //fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian
       );
  fmi.SetDataSetTransferSyntax(
      gdcm::TransferSyntax::ExplicitVRLittleEndian
       );

  w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.55 GenerateDICOMDIR.cs

This is a C# example on how to use gdcm::DICOMDIRGenerator

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Simple C# example to show how to use DICOMDIRGenerator
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/GenerateDICOMDIR.exe path output_filename
 */
using System;
using gdcm;

public class GenerateDICOMDIR
{
  public static int Main(string[] args)
    {
    string directory = args[0];
    string outfilename = args[1];
```

```
    Directory d = new Directory();
    uint nfiles = d.Load( directory, true );
    if(nfiles == 0) return 1;
    //System.Console.WriteLine( "Files:\n" + d.toString() );

    // Implement fast path ?
    // Scanner s = new Scanner();

    string descriptor = "My_Descriptor";
    FilenamesType filenames = d.GetFilenames();

    gdcm.DICOMDIRGenerator gen = new DICOMDIRGenerator();
    gen.SetFilenames( filenames );
    gen.SetDescriptor( descriptor );
    if( !gen.Generate() )
      {
      return 1;
      }

    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "GenerateDICOMDIR
      " );
    gdcm.Writer writer = new Writer();
    writer.SetFile( gen.GetFile() );
    writer.SetFileName( outfilename );
    if( !writer.Write() )
      {
      return 1;
      }

    return 0;
    }
}
```

27.56 GenerateRTSTRUCT.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMPolyDataWriter.h"
#include "vtkGDCMPolyDataReader.h"
#include "vtkPolyData.h"
#include "vtkPolyDataReader.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRTStructSetProperties.h"
#include "vtkStringArray.h"
#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkImageData.h"

#include "gdcmDirectoryHelper.h"

using namespace gdcm;

/*
 * Full application which ... RTSTUCT
 */
```

```
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr << argv[0] << " directory-with-rtstruct-and-ct-images\n";
    return 1;
    }
  std::string theDirName(argv[1]);
  Directory::FilenamesType theRTSeries =
    DirectoryHelper::GetRTStructSeriesUIDs
      (theDirName);
  if (theRTSeries.empty())
    {
    std::cerr << "No RTStructs found for the test, ending." << std::endl;
    return 1;
    }

  Directory::FilenamesType theRTNames =
    DirectoryHelper::GetFilenamesFromSeriesUIDs
      (theDirName, theRTSeries[0]);

  vtkGDCMPolyDataReader * reader =
      vtkGDCMPolyDataReader::New();
  reader->SetFileName( theRTNames[0].c_str() );
  reader->Update();

  //std::cout << reader->GetMedicalImageProperties()->GetStudyDate() <<
       std::endl;

  vtkGDCMPolyDataWriter * writer =
      vtkGDCMPolyDataWriter::New();
  int numMasks = reader->GetNumberOfOutputPorts() + 1;//add a blank one in
  writer->SetNumberOfInputPorts( numMasks );
  writer->SetFileName( std::string(theDirName + "/" + "GDCMTestRTStruct." +
      theRTSeries[0] + ".dcm").c_str());
  writer->SetMedicalImageProperties( reader->
      GetMedicalImageProperties() );
  //this line is cheating, we won't have the same stuff, and may not have a
       struct
  //to start with.
  //have to go back to the original data to reconstruct the
       RTStructureSetProperties
  //writer->SetRTStructSetProperties( reader->GetRTStructSetProperties() );
  //writer->Write();

  //loop through the outputs in order to write them out as if they had been
       created and appended
  vtkStringArray* roiNames = vtkStringArray::New();
  vtkStringArray* roiAlgorithms = vtkStringArray::New();
  vtkStringArray* roiTypes = vtkStringArray::New();
  roiNames->SetNumberOfValues(numMasks);
  roiAlgorithms->SetNumberOfValues(numMasks);
  roiTypes->SetNumberOfValues(numMasks);
  vtkAppendPolyData* append = vtkAppendPolyData::New();
  for (int i = 0; i < reader->GetNumberOfOutputPorts(); ++i)
    {
    writer->SetInput(i, reader->GetOutput(i));
    append->AddInput(reader->GetOutput(i));
    std::string theString = reader->GetRTStructSetProperties()->
      GetStructureSetROIName(i);
    roiNames->InsertValue(i, theString);
    theString = reader->GetRTStructSetProperties()->
      GetStructureSetROIGenerationAlgorithm(i);
    roiAlgorithms->InsertValue(i, theString);
    theString = reader->GetRTStructSetProperties()->
      GetStructureSetRTROIInterpretedType(i);
    roiTypes->InsertValue(i, theString);
    }
  //ok, now we'll add a blank organ
  //the blank organ is to test to ensure that blank organs work; there have
       been crash reports
  vtkPolyData* blank = vtkPolyData::New();
  writer->SetInput(numMasks-1, blank);
  roiNames->InsertValue(numMasks-1, "blank");
  roiAlgorithms->InsertValue(numMasks-1, "blank");
  roiTypes->InsertValue(numMasks-1, "ORGAN");

  vtkRTStructSetProperties* theProperties =
      vtkRTStructSetProperties::New();
  writer->SetRTStructSetProperties(theProperties);
  writer->InitializeRTStructSet(theDirName,
```

```
    reader->GetRTStructSetProperties()->GetStructureSetLabel(),
    reader->GetRTStructSetProperties()->GetStructureSetName(),
    roiNames, roiAlgorithms, roiTypes);

  writer->SetRTStructSetProperties(theProperties);
  writer->Write();

  // print reader output:
  reader->Print( std::cout );
  // print first output:
  reader->GetOutput()->Print( std::cout );

  // Now we'll look at it.
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  cubeMapper->SetInput( append->GetOutput());
  cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();

  vtkRenderer *renderer = vtkRenderer::New();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  renWin->AddRenderer(renderer);

  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renWin);

  renderer->AddActor(cubeActor);
  renderer->ResetCamera();
  renderer->SetBackground(1,1,1);

  renWin->SetSize(300,300);

  renWin->Render();
  iren->Start();

  reader->Delete();
  append->Delete();
  cubeMapper->Delete();
  cubeActor->Delete();
  renderer->Delete();
  renWin->Delete();
  iren->Delete();
  roiNames->Delete();
  roiTypes->Delete();
  theProperties->Delete();
  roiAlgorithms->Delete();
  blank->Delete();

  writer->Delete();

  return 0;
}
```

27.57 GenerateStandardSOPClasses.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 */

#include "gdcmDefs.h"
#include "gdcmUIDs.h"
#include "gdcmGlobal.h"
#include "gdcmMediaStorage.h"
```

```
#include "gdcmSOPClassUIDToIOD.h"

int main(int argc, char *argv[])
{

  using gdcm::MediaStorage;
  gdcm::Global& g = gdcm::Global::GetInstance
      ();
  if( !g.LoadResourcesFiles() )
    {
    std::cerr << "Could not LoadResourcesFiles" << std::endl;
    return 1;
    }

  const gdcm::Defs &defs = g.GetDefs();

  int ret = 0;

  //std::cout << "Table B.5-1 STANDARD SOP CLASSES" << std::endl;
  std::cout << "SOP Class Name,SOP Class UID,IOD Specification (defined in PS
       3.3)" << std::endl;


  gdcm::MediaStorage::MSType mst;
  for ( mst = gdcm::MediaStorage::MediaStorageDirectoryStorage
      ; mst < gdcm::MediaStorage::MS_END;
    mst = (gdcm::MediaStorage::MSType)(mst + 1) )
    {
    const char *iod = defs.GetIODNameFromMediaStorage
      (mst);
    gdcm::UIDs uid;
    uid.SetFromUID( gdcm::MediaStorage::GetMSString
      (mst) /*mst.GetString()*/ );
    if( iod )
      {
      const char *iod_ref = gdcm::SOPClassUIDToIOD::GetIOD
      (uid);
      if( iod_ref )
        {
        std::string iod_ref_str = iod_ref;
        //iod_ref_str += " IOD Modules";
        //if( iod_ref_str != iod )
          {
          //std::cout << "UID: " << uid << "   ";
          std::cout << '"' << uid.GetName() << '"' <<"," << '"' <<uid.
      GetString() << '"' <<"," << '"' <<iod << '"' << std::endl;
          //std::cout << "Incompatible IODs: [" << iod << "] versus ref= [" <<
       iod_ref_str << "]" << std::endl;
          ++ret;
          }
        }
      }
    }


  return 0;
}
```

27.58 GenFakeIdentifyFile.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"
```

```
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmDicts.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"

#include <cstdlib>
#include <cstring>

gdcm::DataElement CreateFakeElement(gdcm::Tag const &
      tag, bool toremove)
{
  static const gdcm::Global &g = gdcm::Global::GetInstance
      ();
  static const gdcm::Dicts &dicts = g.GetDicts();
  static const gdcm::Dict &pubdict = dicts.GetPublicDict
      ();
  static size_t countglobal = 0;
  static std::vector<gdcm::Tag> balcptags =
    gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
      ();
  size_t count = countglobal % balcptags.size();

  const gdcm::DictEntry &dictentry = pubdict.GetDictEntry
      (tag);

  gdcm::DataElement de;
  de.SetTag( tag );
  using gdcm::VR;
  const VR &vr = dictentry.GetVR();
  //if( vr != VR::INVALID )
  if( vr.IsDual() )
    {
    if( vr == VR::US_SS )
      {
      de.SetVR( VR::US );
      }
    else if( vr == VR::US_SS_OW )
      {
      de.SetVR( VR::OW );
      }
    else if( vr == VR::OB_OW )
      {
      de.SetVR( VR::OB );
      }
    }
  else
    {
    de.SetVR( vr );
    }
  const char str[] = "BasicApplicationLevelConfidentialityProfileAttributes";
  const char safe[] = "This is safe to keep";
  if( de.GetVR() != VR::SQ )
    {
    if( toremove )
      de.SetByteValue( str, strlen(str) );
    else
      de.SetByteValue( safe, strlen(safe) );
    }
  else
    {
    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    // Insert sequence into data set
    assert(de.GetVR() == gdcm::VR::SQ );
    gdcm::SmartPointer<gdcm::SequenceOfItems>
       sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();
    de.SetValue(*sq);
    de.SetVLToUndefined();
    //ds.Insert(de);
```

```
    if( !toremove )
      {
      nds.Insert( CreateFakeElement( balcptags[count], true ) );
      countglobal++;
      }
    else
      {
      gdcm::Attribute<0x0008,0x0000> at1 = { 0 };
       // This element has no reason to be 'anonymized'...
      nds.Insert( at1.GetAsDataElement() );
      gdcm::Attribute<0x000a,0x0000> at2 = { 0 };
      nds.Insert( at2.GetAsDataElement() );
      }
    sq->AddItem(it);
    }
  return de;
}

/*
 */
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr << argv[0] << " output.dcm" << std::endl;
    return 1;
    }
  using gdcm::Tag;
  using gdcm::VR;
  const char *outfilename = argv[1];

  std::vector<gdcm::Tag> balcptags =
    gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
      ();

  gdcm::Writer w;
  gdcm::File &f = w.GetFile();
  gdcm::DataSet &ds = f.GetDataSet();

  // Add attribute that need to be anonymized:
  std::vector<gdcm::Tag>::const_iterator it = balcptags.begin();
  for(; it != balcptags.end(); ++it)
    {
    ds.Insert( CreateFakeElement( *it, true ) );
    }

  // Add attribute that do NOT need to be anonymized:
  static const gdcm::Global &g = gdcm::Global::GetInstance
      ();
  static const gdcm::Dicts &dicts = g.GetDicts();
  static const gdcm::Dict &pubdict = dicts.GetPublicDict
      ();

  using gdcm::Dict;
  Dict::ConstIterator dictit = pubdict.Begin();
  for(; dictit != pubdict.End(); ++dictit)
    {
    const gdcm::Tag &dicttag = dictit->first;
    if( dicttag == Tag(0x6e65,0x6146) ) break;
    //const gdcm::DictEntry &dictentry = dictit->second;
    ds.Insert( CreateFakeElement( dicttag, false ) );
    }
  ds.Remove( gdcm::Tag(0x400,0x500) );
  ds.Remove( gdcm::Tag(0x12,0x62) );
  ds.Remove( gdcm::Tag(0x12,0x63) );

  // Make sure to override any UID stuff
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  //ds.Insert( de );
  ds.Replace( de );

  de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage
      );
  de.SetByteValue( ms.GetString(), strlen(ms.GetString()));
```

```
  ds.Replace( de ); // replace !

  gdcm::FileMetaInformation &fmi = f.GetHeader
      ();
  //fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian
       );
  fmi.SetDataSetTransferSyntax(
      gdcm::TransferSyntax::ExplicitVRLittleEndian
       );

  w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.59 GenFakeImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmFileDerivation.h"
#include "gdcmUIDGenerator.h"
//#include "gdcmImageChangePhotometricInterpretation.h"

/*
 * This example shows two things:
 * 1. How to create an image ex-nihilo
 * 2. How to use the gdcm.FileDerivation filter. This filter is meant to create
       "DERIVED" image
 * object. FileDerivation has a simple API where you can reference *all* the
       input image that have been
 * used to generate the image. The API also allows user to specify the purpose
       of reference (see CID 7202,
 * PS 3.16 - 2008), and the image derivation type (CID 7203, PS 3.16 - 2008).
 */
int main(int, char *[])
{
  // Step 1: Fake Image
  gdcm::SmartPointer<gdcm::Image> im = new
      gdcm::Image;

  char * buffer = new char[ 256 * 256 * 3];
  char * p = buffer;
  int b = 128;
  int ybr[3];
  int ybr2[3];
  int rgb[3];

  for(int r = 0; r < 256; ++r)
    for(int g = 0; g < 256; ++g)
      //for(int b = 0; b < 256; ++b)
      {
      rgb[0] = r;
      rgb[1] = g;
      rgb[1] = 128;
      rgb[2] = b;
      ybr[0] = r;
      ybr[1] = g;
      ybr[1] = 128;
      ybr[2] = b;
```

```
    ybr2[0] = r;
    ybr2[1] = g;
    ybr2[1] = 128;
    ybr2[2] = b;
    //gdcm::ImageChangePhotometricInterpretation::YBR2RGB(rgb, ybr);
    //gdcm::ImageChangePhotometricInterpretation::RGB2YBR(ybr2, rgb);
    *p++ = ybr2[0];
    *p++ = ybr2[1];
    *p++ = ybr2[2];
    }

im->SetNumberOfDimensions( 2 );
im->SetDimension(0, 256 );
im->SetDimension(1, 256 );

im->GetPixelFormat().SetSamplesPerPixel(3);
//im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::RGB );
im->SetPhotometricInterpretation(
    gdcm::PhotometricInterpretation::YBR_FULL
     );

unsigned long l = im->GetBufferLength();
if( l != 256 * 256 * 3 )
  {
  return 1;
  }
gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0
    x0010) );
pixeldata.SetByteValue( buffer, l );
delete[] buffer;
im->SetDataElement( pixeldata );

gdcm::UIDGenerator uid; // helper for uid generation

gdcm::SmartPointer<gdcm::File> file = new
    gdcm::File; // empty file

// Step 2: DERIVED object
gdcm::FileDerivation fd;
// For the pupose of this execise we will pretend that this image is
     referencing
// two source image (we need to generate fake UID for that).
const char ReferencedSOPClassUID[] = "1.2.840.10008.5.1.4.1.1.7"; //
     Secondary Capture
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

// Again for the purpose of the exercise we will pretend that the image is a
// multiplanar reformat (MPR):
// CID 7202 Source Image Purposes of Reference
// {"DCM",121322,"Source image for image processing operation"},
fd.SetPurposeOfReferenceCodeSequenceCodeValue
    ( 121322 );
// CID 7203 Image Derivation
// { "DCM",113072,"Multiplanar reformatting" },
fd.SetDerivationCodeSequenceCodeValue( 1130
    72 );
fd.SetFile( *file );
// If all Code Value are ok the filter will execute properly
if( !fd.Derive() )
  {
  std::cerr << "Sorry could not derive using input info" << std::endl;
  return 1;
  }

// We pass both :
// 1. the fake generated image
// 2. the 'DERIVED' dataset object
// to the writer.
gdcm::ImageWriter w;
w.SetImage( *im );
w.SetFile( fd.GetFile() );

// Set the filename:
w.SetFileName( "ybr2.dcm" );
if( !w.Write() )
  {
  return 1;
  }
```

```
  return 0;
}
```

27.60 GenLongSeqs.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

/*
 * This example is used to generate the file:
 *
 *
 * There is a flaw in the DICOM design were it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most case, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * We need to make sure that we can store numerous Item in a SQ
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  // Create a Sequence
  gdcm::SmartPointer<gdcm::SequenceOfItems>
       sq = new gdcm::SequenceOfItems();
  sq->SetLengthToUndefined();

  const char owner_str[] = "GDCM CONFORMANCE TESTS";
  gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
  owner.SetByteValue(owner_str, strlen(owner_str));
  owner.SetVR( gdcm::VR::LO );

  size_t nitems = 1000;
  nitems += std::numeric_limits<uint32_t>::max();
  for(unsigned int idx = 0; idx < nitems; ++idx)
    {
    // Create a dataelement
    //gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
    //de.SetByteValue(ptr, ptr_len);
    //de.SetVR( gdcm::VR::OB );
```

```
    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    //gdcm::DataSet &nds = it.GetNestedDataSet();
    //nds.Insert(owner);
    //nds.Insert(de);

    sq->AddItem(it);
    }

  // Insert sequence into data set
  gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();

  ds.Insert(owner);
  ds.Insert(des);

  gdcm::Writer w;
  w.SetFile( file );
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.61 GenSeqs.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

/*
 * This example is used to generate the file:
 *
 * gdcmConformanceTests/
      SequenceWithUndefinedLengthNotConvertibleToDefinedLength.dcm
 *
 * There is a flaw in the DICOM design were it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most case, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * Deflated syntax was used in this case since this synthetic example can be
 * nicely compressed using this transfer syntax.
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
```

```
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  //const unsigned int nitems = 1000;
  const unsigned int ptr_len = 42; /*94967296 / nitems; */
  //assert( ptr_len == 42949672 );
  char *ptr = new char[ptr_len];
  memset(ptr,0,ptr_len);

  // Create a Sequence
  gdcm::SmartPointer<gdcm::SequenceOfItems>
       sq = new gdcm::SequenceOfItems();
  sq->SetLengthToUndefined();

  const char owner_str[] = "GDCM CONFORMANCE TESTS";
  gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
  owner.SetByteValue(owner_str, strlen(owner_str));
  owner.SetVR( gdcm::VR::LO );

  for(unsigned int idx = 0; idx < 10/* nitems*/; ++idx)
    {
    // Create a dataelement
    gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) )
      ;
    de.SetByteValue(ptr, ptr_len);
    de.SetVR( gdcm::VR::OB );

    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert(owner);
    nds.Insert(de);

    sq->AddItem(it);
    }

  // Insert sequence into data set
  gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();

  ds.Insert(owner);
  ds.Insert(des);

  gdcm::Writer w;
  w.SetFile( file );
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.62 GetArray.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/GetArray.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class GetArray
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    Image image = reader.GetImage();

    PixelFormat pixeltype = image.GetPixelFormat();

    if( image.GetNumberOfDimensions() != 2 )
      {
      // For the purpose of the test, exit early on
      return 1;
      }
    uint dimx = image.GetDimension(0);
    uint dimy = image.GetDimension(1);
    uint npixels = dimx * dimy;
    //LookupTable lut = image.GetLUT();
    //uint rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
    //byte[] rbuf = new byte[ rl ];
    //uint rl2 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
    //assert rl == rl2;

    //byte[] str1 = new byte[ image.GetBufferLength()];
    //image.GetBuffer( str1 );
    if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT8 )
      {
      System.Console.WriteLine( "Processing UINT8 image type" );
      byte[] str1 = new byte[ npixels ];
      image.GetArray( str1 );
      }
    else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.INT16 )
      {
      System.Console.WriteLine( "Processing INT16 image type" );
      short[] str1 = new short[ npixels ];
      image.GetArray( str1 );
      }
    else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT16 )
      {
      System.Console.WriteLine( "Processing UINT16 image type" );
      ushort[] str1 = new ushort[ npixels ];
      image.GetArray( str1 );
      }
    else
      {
      //System.Console.WriteLine( "Default (unhandled pixel format): " +
       pixeltype.toString() );
      System.Console.WriteLine( "Default (unhandled pixel format): " +
      pixeltype.GetScalarTypeAsString() );
      // Get bytes
      byte[] str1 = new byte[ image.GetBufferLength()];
      image.GetBuffer( str1 );
      }

    return 0;
    }
}
```

27.63 GetJPEGSamplePrecision.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example is a little helper to detect the famous SIEMENS JPEG lossless
       compressed image
 * where DICOM is declared as:
 *
 *  (0028,0100) US 16                                                # 2,1
       Bits Allocated
 *  (0028,0101) US 12                                                # 2,1
       Bits Stored
 *  (0028,0102) US 11                                                # 2,1
       High Bit
 *  (0028,0103) US 0                                                 # 2,1
       Pixel Representation
 *
 * But where JPEG is:
 *
 *        JPEG_SOF_Parameters:
 *                SamplePrecision = 16
 *                nLines = 192
 *                nSamplesPerLine = 192
 *                nComponentsInFrame = 1
 *                component 0
 *                        ComponentIdentifier = 1
 *                        HorizontalSamplingFactor = 1
 *                        VerticalSamplingFactor = 1
 *                        QuantizationTableDestinationSelector = 0
 *
 *
 * This case is valid. One simply has to use the 16bits jpeg decoder to decode
       the 12bits stored image.
 * This used to be an issue in GDCM 1.2.x (fixed in GDCM 1.2.5)
 *
 * The main return 0 (no error) when the file read is actually a potential
       problem. At the end of the main
 * function, the jpeg stream is stored in the filename specified as second
       argument
 */

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmJPEGCodec.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.jpg" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
    }

  // The output of gdcm::Reader is a gdcm::File
  const gdcm::File &file = reader.GetFile();
```

```
  const gdcm::Image &image = reader.GetImage();

  const gdcm::TransferSyntax &ts = file.GetHeader(
      ).GetDataSetTransferSyntax();

  if( ts != gdcm::TransferSyntax::JPEGLosslessProcess14
       && ts != gdcm::TransferSyntax::JPEGLosslessProcess14_1
       )
    {
    std::cerr << "Input is not a lossless JPEG" << std::endl;
    return 1;
    }

  // the dataset is the the set of element we are interested in:
  const gdcm::DataSet &ds = file.GetDataSet();

  const gdcm::Tag rawTag(0x7fe0, 0x0010); // Default to Pixel Data
  const gdcm::DataElement& pdde = ds.GetDataElement
      ( rawTag );
  const gdcm::SequenceOfFragments *sf = pdde.
      GetSequenceOfFragments();
  if( sf )
    {
    std::ofstream output(outfilename, std::ios::binary);
    sf->WriteBuffer(output);
    }
  else
    {
    std::cerr << "Error" << std::endl;
    return 1;
    }

  gdcm::JPEGCodec jpeg;
  std::ifstream is(outfilename);
  gdcm::PixelFormat pf ( gdcm::PixelFormat::UINT8
       ); // let's pretend it's a 8bits jpeg
  jpeg.SetPixelFormat( pf );
  gdcm::TransferSyntax ts_jpg;
  bool b = jpeg.GetHeaderInfo( is, ts_jpg );
  if( !b )
    {
    return 1;
    }

  //jpeg.Print( std::cout );
  if( jpeg.GetPixelFormat().GetBitsAllocated() !=
       image.GetPixelFormat().GetBitsAllocated()
   || jpeg.GetPixelFormat().GetBitsStored() != image
      .GetPixelFormat().GetBitsStored() )
    {
    std::cerr << "There is a mismatch in between DICOM declared Pixel Format
       and Sample Precision used in the JPEG stream" << std::endl;
    return 0;
    }

  std::cout << jpeg.GetPixelFormat() << std::endl;
  std::cout << image.GetPixelFormat() << std::endl;

  return 1;
}
```

27.64 GetPortionCSAHeader.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################
```

```
"""
Usage:

 python GetPortionCSAHeader.py input.dcm

Footnote:
  SIEMENS is not publishing any information on the CSA header. So any info
       extracted
  is at your own risk.
"""

import sys
import gdcm

if __name__ == "__main__":

  file = sys.argv[1]

  r = gdcm.Reader()
  r.SetFileName( file )
  if not r.Read():
    sys.exit(1)

  ds = r.GetFile().GetDataSet()
  csa_t1 = gdcm.CSAHeader()
  csa_t2 = gdcm.CSAHeader()
  #print csa
  t1 = csa_t1.GetCSAImageHeaderInfoTag();
  print t1
  t2 = csa_t2.GetCSASeriesHeaderInfoTag();
  print t2
  # Let's do it for t1:
  if ds.FindDataElement( t1 ):
    csa_t1.LoadFromDataElement( ds.GetDataElement( t1 ) )
    print csa_t1

  # Now let's pretend we are only interested in B_value and
       DiffusionGradientDirection entries:
  bvalues = csa_t1.GetCSAElementByName( "B_value" ) # WARNING: it is case
       sensitive !
  print bvalues

  diffgraddir = csa_t1.GetCSAElementByName( "DiffusionGradientDirection" ) #
       WARNING: it is case sensitive !
  print diffgraddir

  # repeat for t2 if you like it:
  if ds.FindDataElement( t2 ):
    csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
    # print csa_t2

  gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
  print gdt

  bv = gdt.GetByteValue();
  #print bv
  str = bv.GetPointer()
  print str.split("\\")
```

27.65 GetSequenceUltrasound.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmAttribute.h"
```

```
bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min,
      unsigned int* X_max, unsigned int* Y_max );

int main(int argc, char* argv[] )
{
  // Controllo del numero di argomenti introdotti da riga di comando
  if( argc < 2 )
    {
    std::cerr << "Usage: " << std::endl;
    std::cerr << argv[0] << " inputImageFile  " << std::endl;
    return EXIT_FAILURE;
    }


  unsigned int x_min = 1;
  unsigned int y_min = 1;
  unsigned int x_max = 1;
  unsigned int y_max = 1;

  if( Region ( argv[1], &x_min, &y_min, &x_max, &y_max ) )
    {
    std::cout << "x_min = " << x_min << std::endl;
    std::cout << "y_min = " << y_min << std::endl;
    std::cout << "x_max = " << x_max << std::endl;
    std::cout << "y_max = " << y_max << std::endl;
    }

  else
    {
    std::cout << "no\n";
    }

}

bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min,
      unsigned int* X_max, unsigned int* Y_max )
{
  gdcm::Reader reader;
  reader.SetFileName( nomefile );
  if( !reader.Read() )
    {
    std::cerr << "Could not read: " << nomefile << std::endl;
    return false;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  gdcm::Tag tsqur(0x0018,0x6011);
  if( !ds.FindDataElement( tsqur ) )
    {
    return false;
    }

  const gdcm::DataElement &squr= ds.GetDataElement
      ( tsqur );
  //std::cout << squr << std::endl;
  const gdcm::SequenceOfItems *sqi = squr.GetValueAsSQ
      ();
  if( !sqi || !sqi->GetNumberOfItems() )
    {
    return false;
    }
  //std::cout << sqi << std::endl;

  const gdcm::Item & item = sqi->GetItem(1);
  //std::cout << item << std::endl;
  const gdcm::DataSet& nestedds = item.GetNestedDataSet
      ();
  //std::cout << nestedds << std::endl;

  gdcm::Tag tX0(0x0018,0x6018);
  gdcm::Tag tY0(0x0018,0x601a);
  gdcm::Tag tX1(0x0018,0x601c);
  gdcm::Tag tY1(0x0018,0x601e);

  if( (!nestedds.FindDataElement( tX0 ))||(!nestedds.
      FindDataElement( tY0 ))||(!nestedds.FindDataElement
      ( tX1 ))||(!nestedds.FindDataElement( tY1 )) )
    {
    return false;
```

```
    }

  const gdcm::DataElement& deX0 = nestedds.GetDataElement
      ( tX0 );
  const gdcm::DataElement& deY0 = nestedds.GetDataElement
      ( tY0 );
  const gdcm::DataElement& deX1 = nestedds.GetDataElement
      ( tX1 );
  const gdcm::DataElement& deY1 = nestedds.GetDataElement
      ( tY1 );
  //std::cout << deX0 << std::endl << deY0 << std::endl << deX1 << std::endl <<
       deY1 << std::endl;

  //const gdcm::ByteValue *bvX0 = deX0.GetByteValue();
  //const gdcm::ByteValue *bvY0 = deY0.GetByteValue();
  //const gdcm::ByteValue *bvX1 = deX1.GetByteValue();
  //const gdcm::ByteValue *bvY1 = deY1.GetByteValue();
  //std::cout << bvX0 << std::endl << bvY0 << std::endl << bvX1 << std::endl <<
       bvY1 << std::endl;

  gdcm::Attribute<0x0018,0x6018> atX0;
  gdcm::Attribute<0x0018,0x601a> atY0;
  gdcm::Attribute<0x0018,0x601c> atX1;
  gdcm::Attribute<0x0018,0x601e> atY1;
  atX0.SetFromDataElement( deX0 );
  atY0.SetFromDataElement( deY0 );
  atX1.SetFromDataElement( deX1 );
  atY1.SetFromDataElement( deY1 );
  uint32_t X0 = atX0.GetValue();
  uint32_t Y0 = atY0.GetValue();
  uint32_t X1 = atX1.GetValue();
  uint32_t Y1 = atY1.GetValue();
  std::cout << X0 << std::endl << Y0 << std::endl << X1 << std::endl << Y1 <<
      std::endl;

  *X_min = static_cast<unsigned int>(X0);
  *Y_min = static_cast<unsigned int>(Y0);
  *X_max = static_cast<unsigned int>(X1);
  *Y_max = static_cast<unsigned int>(Y1);

  //std::cout << "X_min = " << *X_min << std::endl;
  //std::cout << "Y_min = " << *Y_min << std::endl;
  //std::cout << "X_max = " << *X_max << std::endl;
  //std::cout << "Y_max = " << *Y_max << std::endl;

  return true;
}
```

27.66 GetSubSequenceData.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

/*
 * This example will extract the Movie from the private group of
```

```
 * GEMS_Ultrasound_MovieGroup_001 See Attribute
 * (7fe1,60,GEMS_Ultrasound_MovieGroup_001)
 *
 * The output file will be stored in `outvid.dcm` as
 * MultiframeGrayscaleByteSecondaryCaptureImageStorage
 */
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  reader.Read();

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");

  if( !ds.FindDataElement( tseq ) ) return 1;
  const DataElement& seq = ds.GetDataElement( tseq );

  SmartPointer<SequenceOfItems> sqi = seq.
      GetValueAsSQ();
  assert( sqi->GetNumberOfItems() == 1 );
  Item &item = sqi->GetItem(1);
  DataSet &subds = item.GetNestedDataSet();

  const PrivateTag tseq1(0x7fe1,0x10,"GEMS_Ultrasound_MovieGroup_001"
      );

  if( !subds.FindDataElement( tseq1 ) ) return 1;
  const DataElement& seq1 = subds.GetDataElement( tseq1 );

  SmartPointer<SequenceOfItems> sqi2 = seq1.
      GetValueAsSQ();
  //int n = sqi2->GetNumberOfItems();
  int index = 1;
  Item &item2 = sqi2->GetItem(index);
  DataSet &subds2 = item2.GetNestedDataSet();

  const PrivateTag tseq2(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001"
      );

  if( !subds2.FindDataElement( tseq2 ) ) return 1;
  const DataElement& seq2 = subds2.GetDataElement( tseq2 );

//    std::cout << seq2 << std::endl;

  SmartPointer<SequenceOfItems> sqi3 = seq2.
      GetValueAsSQ();
  int ni3 = sqi3->GetNumberOfItems();
  assert( sqi3->GetNumberOfItems() >= 1 );
  Item &item3 = sqi3->GetItem(1);
  DataSet &subds3 = item3.GetNestedDataSet();

  const PrivateTag tseq6(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001"
      );
  if( !subds3.FindDataElement( tseq6 ) ) return 1;
  const DataElement& seq6 = subds3.GetDataElement( tseq6 );
  SmartPointer<SequenceOfItems> sqi6 = seq6.
      GetValueAsSQ();
  int ni6= sqi6->GetNumberOfItems();
  assert( sqi6->GetNumberOfItems() >= 1 );
  const PrivateTag tseq7(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001"
      );
  int dimx, dimy;
  for( int i6 = 1; i6 <= ni6; ++i6 )
    {
    Item &item6 = sqi6->GetItem(i6);
    DataSet &subds6 = item6.GetNestedDataSet();

    if( subds6.FindDataElement( tseq7 ) )
      {
      Element<VR::SL, VM::VM4> el;
      el.SetFromDataElement( subds6.GetDataElement
      ( tseq7 ) );
      std::cout << "El= " << el.GetValue() << std::endl;
      dimx = el.GetValue(0);
      dimy = el.GetValue(1);
      }
```

```
    }

  const PrivateTag tseq3(0x7fe1,0x36,"GEMS_Ultrasound_MovieGroup_001"
      );
  if( !subds3.FindDataElement( tseq3 ) ) return 1;
  const DataElement& seq3 = subds3.GetDataElement( tseq3 );

//    std::cout << seq3 << std::endl;

  SmartPointer<SequenceOfItems> sqi4 = seq3.
      GetValueAsSQ();
  int ni4= sqi4->GetNumberOfItems();
  assert( sqi4->GetNumberOfItems() >= 1 );
  const PrivateTag tseq8(0x7fe1,0x37,"GEMS_Ultrasound_MovieGroup_001"
      );
  const PrivateTag tseq4(0x7fe1,0x43,"GEMS_Ultrasound_MovieGroup_001"
      );
  const PrivateTag tseq5(0x7fe1,0x60,"GEMS_Ultrasound_MovieGroup_001"
      );

  std::vector<char> imbuffer;
  int dimz = 0;
  for( int i4 = 1; i4 <= ni4; ++i4 )
    {
    Item &item4 = sqi4->GetItem(i4);
    DataSet &subds4 = item4.GetNestedDataSet();

    if( !subds4.FindDataElement( tseq8 ) ) return 1;
    const DataElement& de8 = subds4.GetDataElement(
      tseq8 );
    Element<VR::UL,VM::VM1> ldimz;
    ldimz.SetFromDataElement( de8 );
    dimz += ldimz.GetValue();
    if( !subds4.FindDataElement( tseq4 ) ) return 1;
    const DataElement& seq4 = subds4.GetDataElement(
      tseq4 );
    if( !subds4.FindDataElement( tseq5 ) ) return 1;
    const DataElement& seq5 = subds4.GetDataElement(
      tseq5 );

    //    std::cout << seq4 << std::endl;
    //    std::cout << seq5 << std::endl;

    const ByteValue *bv4 = seq4.GetByteValue();
#if 0
      {
      std::ofstream out( "/tmp/mo4" );
      out.write( bv4->GetPointer(), bv4->GetLength());
      out.close();
      }
#endif
  const ByteValue *bv5 = seq5.GetByteValue();
#if 0
    {
    std::ofstream out( "/tmp/mo5" );
    out.write( bv5->GetPointer(), bv5->GetLength());
    out.close();
    }
#endif

    std::cout << bv5->GetLength() << std::endl;
    imbuffer.insert( imbuffer.begin(), bv5->GetPointer(), bv5->
      GetPointer() + bv5->GetLength() );
    }
  DataElement fakedata;
  fakedata.SetByteValue( &imbuffer[0], imbuffer.size() );


  gdcm::SmartPointer<gdcm::Image> im = new
      gdcm::Image;
  im->SetNumberOfDimensions( 3 );

  im->SetDimension(0, dimx );
  im->SetDimension(1, dimy );
  im->SetDimension(2, dimz );
  size_t l1 = imbuffer.size();
  size_t l2 = im->GetBufferLength();
  assert( im->GetBufferLength() == imbuffer.size() );
  im->SetPhotometricInterpretation(
      gdcm::PhotometricInterpretation::MONOCHROME2 );
```

```
  im->SetDataElement( fakedata );

  gdcm::ImageWriter w;
  w.SetImage( *im );
  DataSet &dataset = w.GetFile().GetDataSet();

  gdcm::UIDGenerator uid;
  gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance
       UID
  de.SetVR( VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  //ds.Insert( de );
  dataset.Replace( de );

  de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( VR::UI );
  gdcm::MediaStorage ms(
    gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage
       );
  de.SetByteValue( ms.GetString(), strlen(ms.GetString
      ()));
  dataset.Replace( de ); // replace !

  w.SetFileName( "outvid.dcm" );
  if( !w.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.67 headsq2dcm.py

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

"""
Usage:
 python headsq2dcm.py -D /path/to/VTKData
"""

import vtk
import vtkgdcm
from vtk.util.misc import vtkGetDataRoot
VTK_DATA_ROOT = vtkGetDataRoot()

reader = vtk.vtkVolume16Reader()
reader.SetDataDimensions(64, 64)
reader.SetDataByteOrderToLittleEndian()
reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter")
reader.SetImageRange(1, 93)
reader.SetDataSpacing(3.2, 3.2, 1.5)

cast = vtk.vtkImageCast()
cast.SetInput( reader.GetOutput() )
cast.SetOutputScalarTypeToUnsignedChar()

# By default this is creating a Multiframe Grayscale Word Secondary Capture
       Image Storage
writer = vtkgdcm.vtkGDCMImageWriter()
writer.SetFileName( "headsq.dcm" )
writer.SetInput( reader.GetOutput() )
# cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
#writer.SetInput( cast.GetOutput() )
```

```
writer.SetFileDimensionality( 3 )
writer.Write()
```

27.68 HelloActiviz.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using vtkgdcm;
using Kitware.VTK;
using System;
using System.Runtime.InteropServices;

/*
 * This example shows how vtkgdcm can be connected to Kitware.VTK Activiz
       product.
 * Three (3) arguments are required:
 * 1. Input DICOM file                      (SWIG)
 * 2. Temporary PNG (intermediate) file     (Activiz)
 * 3. Final DICOM file                      (SWIG)
 *
 * $ MONO_PATH=/home/mmalaterre/Software/
      ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/ LD_LIBRARY_PATH=/home/mmalaterre/Software/
      ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/:/home/mmalaterre/Projects/gdcm/debug-gcc43/bin/ mono ./bin
      /HelloActiviz.exe ~/Creatis/gdcmData/test.acr out.png toto.dcm
 *
 * Footnote:
 * this test originally used vtkBMPWriter / vtkBMPReader combination to store
       intermediate
 * image file, but BMP file are 24bits by default. Instead use PNG format which
       supports seems
 * to be closer to what was expected in this simple test.
 */
public class HelloActiviz
{
  // Does not work with ActiViz.NET-5.4.0.455-Linux-x86_64-Personal
/*
  static void ConnectSWIGToActiviz(Kitware.VTK.vtkImageExport imgin,
       Kitware.VTK.vtkImageImport imgout)
    {
    imgout.SetUpdateInformationCallback(imgin.GetUpdateInformationCallback());
    imgout.SetPipelineModifiedCallback(imgin.GetPipelineModifiedCallback());
    imgout.SetWholeExtentCallback(imgin.GetWholeExtentCallback());
    imgout.SetSpacingCallback(imgin.GetSpacingCallback());
    imgout.SetOriginCallback(imgin.GetOriginCallback());
    imgout.SetScalarTypeCallback(imgin.GetScalarTypeCallback());

       imgout.SetNumberOfComponentsCallback(imgin.GetNumberOfComponentsCallback());

       imgout.SetPropagateUpdateExtentCallback(imgin.GetPropagateUpdateExtentCallback());
    imgout.SetUpdateDataCallback(imgin.GetUpdateDataCallback());
    imgout.SetDataExtentCallback(imgin.GetDataExtentCallback());
    imgout.SetBufferPointerCallback(imgin.GetBufferPointerCallback());
    imgout.SetCallbackUserData(imgin.GetCallbackUserData());
    }
*/

  static Kitware.VTK.vtkImageData ConnectSWIGToActiviz(vtkgdcm.vtkImageData
      imgin)
    {
    HandleRef rawCppThis = imgin.GetCppThis();
    Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis.
      Handle, false, false);
    return imgout;
    }

  static vtkgdcm.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData
```

```
      imgin)
    {
    HandleRef rawCppThis = imgin.GetCppThis();
    vtkgdcm.vtkImageData imgout = new vtkgdcm.vtkImageData( rawCppThis );
    return imgout;
    }


  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];

    // Step 1. Test SWIG -> Activiz
    vtkGDCMImageReader reader = vtkGDCMImageReader
      .New();
    reader.SetFileName( filename );
    //reader.Update(); // DO NOT call Update to check pipeline execution

    Kitware.VTK.vtkImageData imgout = ConnectSWIGToActiviz(reader.GetOutput());

    System.Console.WriteLine( imgout.ToString() ); // not initialized as
       expected

    vtkPNGWriter writer = new vtkPNGWriter();
    writer.SetInput( imgout );
    writer.SetFileName( outfilename );
    writer.Write();

    // Step 2. Test Activiz -> SWIG
    vtkPNGReader bmpreader = new vtkPNGReader();
    bmpreader.SetFileName( outfilename );
    //bmpreader.Update(); // DO NOT update to check pipeline execution

    System.Console.WriteLine( bmpreader.GetOutput().ToString() ); // not
       initialized as expected

    vtkgdcm.vtkImageData imgout2 = ConnectActivizToSWIG(bmpreader.GetOutput());

    System.Console.WriteLine( imgout2.ToString() ); // not initialized as
       expected


    Kitware.VTK.vtkMedicalImageProperties prop = new Kitware.VTK.
      vtkMedicalImageProperties();
    prop.SetModality( "MR" );

    string outfilename2 = args[2];
    vtkGDCMImageWriter writer2 = vtkGDCMImageWriter
      .New();
    writer2.SetMedicalImageProperties( prop.
      CastToActiviz() );
    writer2.SetFileName( outfilename2 );
    writer2.SetInput( imgout2 );
    writer2.Write();

    return 0;
    }
}
```

27.69 HelloActiviz2.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using vtkgdcm;
```

```
/*
 * Usage:
 * export LD_LIBRARY_PATH=/home/mathieu/Perso/gdcm/debug-activiz/bin:/home/
      mathieu/Software/ActiViz.NET-5.4.2.488-Linux-x86_64-Personal/bin
 * export MONO_PATH=/home/mathieu/Software/
      ActiViz.NET-5.4.2.488-Linux-x86_64-Personal/bin
 * $ mono ./bin/HelloActiviz2.exe gdcmData/test.acr bla.png bla2.dcm
 */

/*
 * From the outside view, no-one can detect that object pass to/from
 * vtkGDCMImageWriter/vtkGDCMImageReader are not Activiz object.
 *
 * TODO: Test Command/Observer
 */
public class HelloActiviz2
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];
    string outfilename2 = args[2];

    vtkgdcm.vtkGDCMImageReader reader = new vtkgdcm.vtkGDCMImageReader();
    reader.SetFileName( filename );

    // When calling multiple times creation of C# object from the same C++
       object it triggers a:
//error: potential refcounting error: Duplicate rawCppThis - weak reference
       that is still alive. Attempting to add '0x00b2dc10' again.
//       Allowing new wrapped object to take over table key...
//       Original object should *not* have been destroyed while we still had it
       in our table without notifying us...
    //reader.GetOutput();
    //reader.GetOutput();

    System.Console.WriteLine( reader.ToString() ); // Test the ToString compat
       with Activiz

    vtkGDCMImageWriter writer = new vtkGDCMImageWriter
      ();
    writer.SetInput( reader.GetOutput() );
    writer.SetFileName( outfilename2 );
    writer.Write();

    System.Console.WriteLine( reader.GetOutput().ToString() ); // Test the
       ToString compat with Activiz

    System.Console.WriteLine( writer.ToString() ); // Test the ToString compat
       with Activiz

    vtkPNGWriter pngwriter = new vtkPNGWriter();
    pngwriter.SetInput( reader.GetOutput() );
    pngwriter.SetFileName( outfilename );
    pngwriter.Write();

    // at that point the .Write() should have triggered an Update() on the
       reader:
    if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) //
       MONOCHROME2
      {
      System.Console.WriteLine( "Image is MONOCHROME2" ); //
      }

    vtkPNGReader bmpreader = new vtkPNGReader();
    bmpreader.SetFileName( outfilename );

    vtkMedicalImageProperties prop = new vtkMedicalImageProperties();
    prop.SetModality( "MR" );

    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();

    vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter
      ();
    writer2.SetFileName( outfilename2 );
    writer2.SetDirectionCosines( dircos );
    writer2.SetMedicalImageProperties( prop );
    writer2.SetInput( bmpreader.GetOutput() );
    writer2.Write();
```

```
    return 0;
    }
}
```

27.70 HelloActiviz3.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using vtkgdcm;

/*
 * $ MONO_PATH=/home/mmalaterre/Software/
      ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/ LD_LIBRARY_PATH=/home/mmalaterre/Software/
      ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/:/home/mmalaterre/Projects/gdcm/debug-gcc43/bin/ mono ./bin
      /HelloActiviz3.exe ~/Creatis/gdcmData/test.acr
 *
 */
public class HelloActiviz3
{
  public static int Main(string[] args)
    {
    string filename = args[0];

    vtkGDCMImageReader reader = vtkGDCMImageReader
      .New();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);

    reader.SetFileNames(array);
    reader.Update();

    //System.Console.Write(reader.GetOutput());

    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

    vtkImageViewer2 viewer = vtkImageViewer2.New();
    viewer.SetInput(reader.GetOutput());
    viewer.SetupInteractor(iren);
    viewer.SetSize(600, 600);
    viewer.Render();

    iren.Initialize();
    iren.Start();

    return 0;
    }
}
```

27.71 HelloActiviz4.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
     PURPOSE.  See the above copyright notice for more information.

=====================================================================*/
using Kitware.VTK;
using vtkgdcm;

/*
 * $ MONO_PATH=/home/mmalaterre/Software/
      ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/ LD_LIBRARY_PATH=/home/mmalaterre/Software/
      ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/:/home/mmalaterre/Projects/gdcm/debug-gcc43/bin/ mono ./bin
      /HelloActiviz4.exe ~/Creatis/gdcmData/test.acr
 *
 */
public class HelloActiviz4
{
  public static int Main(string[] args)
    {
    string filename = args[0];

    vtkGDCMImageReader reader = new vtkGDCMImageReader
      ();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);

    reader.SetFileNames(array);
    reader.Update();

    //System.Console.Write(reader.GetOutput());

    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

    vtkImageViewer viewer = vtkImageViewer.New();
    viewer.SetInput(reader.GetOutput());
    viewer.SetupInteractor(iren);
    viewer.SetSize(600, 600);
    viewer.Render();

    iren.Initialize();
    iren.Start();

    return 0;
    }
}
```

27.72 HelloActiviz5.cs

```
/*=====================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=====================================================================*/
using Kitware.VTK;
using vtkgdcm;

// The command line arguments are:
// -I        => run in interactive mode; unless this is used, the program will
//              not allow interaction and exit
// -D <path> => path to the data; the data should be in <path>/Data/

/*
 * $ export MONO_PATH=/home/mmalaterre/Software/
      ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/
 * $ export LD_LIBRARY_PATH=/home/mmalaterre/Software/
      ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/:/home/mmalaterre/Projects/gdcm/debug-gcc43/bin/
 * $ mono ./bin/HelloActiviz5.exe -I
 *
 */
public class HelloActiviz5
{
```

```
  public static int Main(string[] args)
    {
    vtkTesting testHelper = vtkTesting.New();
    for ( int cc = 0; cc < args.Length; cc++ )
      {
      //testHelper.AddArguments(argc,const_cast<const char **>(argv));
      //System.Console.Write( "args: " + args[cc] + "\n" );
      testHelper.AddArgument( args[cc] );
      }
    if ( testHelper.IsFlagSpecified("-D") != 0 )
      {
      string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot
      ();
      if( VTK_DATA_ROOT != null )
        {
        //System.Console.Write( "VTK_DATA_ROOT: " + VTK_DATA_ROOT  + "\n" );
        testHelper.SetDataRoot(VTK_DATA_ROOT);
        testHelper.AddArgument("-D");
        testHelper.AddArgument(VTK_DATA_ROOT);
        }
      }

    string dataRoot = testHelper.GetDataRoot();
    string filename = dataRoot;
    filename += "/Data/mr.001";

    vtkDirectory dir = vtkDirectory.New();
    if( dir.FileIsDirectory( dataRoot ) == 0 )
      {
      filename = vtkGDCMTesting.GetGDCMDataRoot()
      + "/test.acr";
      }
    //System.Console.Write( "dataRoot: " + dataRoot + "\n" );
    System.Console.Write( "filename being used is: " + filename + "\n" );

    vtkGDCMImageReader reader = vtkGDCMImageReader
      .New();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);
    reader.SetFileNames(array);
    reader.Update();

    System.Console.Write(reader.GetOutput());

    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

    vtkRenderer ren1 = vtkRenderer.New();
    vtkRenderWindow renWin = vtkRenderWindow.New();
    renWin.AddRenderer(ren1);

    vtkImageActor actor = vtkImageActor.New();

    vtkImageMapToWindowLevelColors coronalColors =
      vtkImageMapToWindowLevelColors.New();
    coronalColors.SetInput(reader.GetOutput());

    actor.SetInput(coronalColors.GetOutput());

    ren1.AddActor(actor);
    iren.SetRenderWindow(renWin);

    iren.Initialize();

    renWin.Render();

    int retVal = testHelper.IsInteractiveModeSpecified();

    if( retVal != 0 )
      {
      iren.Start();
      }

    return 0;
    }
}
```

27.73 HelloSimple.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/HelloSimple.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java HelloSimple gdcmData/
      012345.002.050.dcm
 */
import gdcm.*;

public class HelloSimple
{
  public static void main(String[] args) throws Exception
    {
    String filename = args[0];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read: " + filename );
      }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();

    System.out.println( ds.toString() );

    System.out.println("Success reading: " + filename );
    }
}
```

27.74 HelloVizWorld.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Basic example for dealing with a DICOM file that contains an Image
 * (read: Pixel Data element)
 */

#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
```

```
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  // Instanciate the image reader:
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
    }
  // If we reach here, we know for sure 2 things:
  // 1. It is a valid DICOM
  // 2. And it contains an Image !

  // The output of superclass gdcm::Reader is a gdcm::File
  //gdcm::File &file = reader.GetFile();

  // The other output of gdcm::ImageReader is a gdcm::Image
  const gdcm::Image &image = reader.GetImage();

  // Let's get some property from the image:
  unsigned int ndim = image.GetNumberOfDimensions();
  // Dimensions of the image:
  const unsigned int *dims = image.GetDimensions();
  // Origin
  const double *origin = image.GetOrigin();
  const gdcm::PhotometricInterpretation &pi =
      image.GetPhotometricInterpretation();
  for(unsigned int i = 0; i < ndim; ++i)
    {
    std::cout << "Dim(" << i << "): " << dims[i] << std::endl;
    }
  for(unsigned int i = 0; i < ndim; ++i)
    {
    std::cout << "Origin(" << i << "): " << origin[i] << std::endl;
    }
  std::cout << "PhotometricInterpretation: " << pi << std::endl;

  // Write the modified DataSet back to disk
  gdcm::ImageWriter writer;
  writer.SetImage( image );
  writer.SetFileName( outfilename );
  //writer.SetFile( file ); // We purposely NOT copy the meta information from
      the input
                          // file, and instead only pass the image
  if( !writer.Write() )
    {
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
    }

  return 0;
}
```

27.75 HelloVTKWorld.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using vtkgdcm;

/*
```

```
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloVTKWorld
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    vtkGDCMImageReader reader = vtkGDCMImageReader
      .New();
    reader.SetFileName( filename );
    reader.Update();

    vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
    System.Console.WriteLine( prop.GetPatientName() ); //

    if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) //
       MONOCHROME2
      {
      System.Console.WriteLine( "Image is MONOCHROME2" ); //
      }

    // Just for fun, invert the direction cosines, output should reflect that:
    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();

    string outfilename = args[1];
    vtkGDCMImageWriter writer = vtkGDCMImageWriter
      .New();
    writer.SetMedicalImageProperties( reader.
      GetMedicalImageProperties() );
    writer.SetDirectionCosines( dircos );
    writer.SetShift( reader.GetShift() );
    writer.SetScale( reader.GetScale() );
    writer.SetImageFormat( reader.GetImageFormat() );
    writer.SetFileName( outfilename );
    //writer.SetInputConnection( reader.GetOutputPort() ); // new
    writer.SetInput( reader.GetOutput() ); // old
    writer.Write();

    return 0;
    }
}
```

27.76 HelloVTKWorld.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// We are required to call the package 'vtk' eventhough I (MM) would have
       prefered
// an import statement along the line of:
// import vtkgdcm.*;
import vtk.*;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac HelloVTKWorld.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/
      jni:. CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java HelloVTKWorld
       gdcmData/012345.002.050.dcm bla.dcm
 *
 */
public class HelloVTKWorld
{
  static {
```

```
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkgdcmJava");
    try {
      System.loadLibrary("vtkRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkHybridJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkVolumeRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkVolumeRendering, skipping...");
    }
  }

  public static void main(String[] args)
    {
    String filename = args[0];
    vtkGDCMImageReader reader = new vtkGDCMImageReader
      ();
    reader.SetFileName( filename );
    reader.Update();

    vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
    System.out.println( prop.GetPatientName() ); //

//    if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) //
      MONOCHROME2
//      {
//      System.out.println( "Image is MONOCHROME2" ); //
//      }

    // Just for fun, invert the direction cosines, output should reflect that:
    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();

    // We need to maintain in sync information stored in
       vtkMedicalImageProperties:
    double[] cosines = new double[6];
    cosines[0] = dircos.GetElement(0,0);
    cosines[1] = dircos.GetElement(1,0);
    cosines[2] = dircos.GetElement(2,0);
    cosines[3] = dircos.GetElement(0,1);
    cosines[4] = dircos.GetElement(1,1);
    cosines[5] = dircos.GetElement(2,1);
    reader.GetMedicalImageProperties().SetDirectionCosine( cosines );

    String outfilename = args[1];
    vtkGDCMImageWriter writer = new vtkGDCMImageWriter
      ();
    writer.SetMedicalImageProperties( reader.
      GetMedicalImageProperties() );
    writer.SetDirectionCosines( dircos );
    writer.SetShift( reader.GetShift() );
    writer.SetScale( reader.GetScale() );
    writer.SetImageFormat( reader.GetImageFormat() );
    writer.SetFileName( outfilename );
    //writer.SetInputConnection( reader.GetOutputPort() ); // new
    writer.SetInput( reader.GetOutput() ); // old
    writer.Write();

    System.out.println("Success reading: " + filename );
    }
}
```

27.77 HelloVTKWorld2.cs

```
/*=========================================================================
```

```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using vtkgdcm;

/*
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloVTKWorld2
{
  public static int Main(string[] args)
    {
    string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot
      ();

    vtkVolume16Reader reader = vtkVolume16Reader.New();
    reader.SetDataDimensions(64, 64);
    reader.SetDataByteOrderToLittleEndian();
    reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter");
    reader.SetImageRange(1, 93);
    reader.SetDataSpacing(3.2, 3.2, 1.5);

    vtkImageCast cast = vtkImageCast.New();
    cast.SetInput( reader.GetOutput() );
    cast.SetOutputScalarTypeToUnsignedChar();

    // By default this is creating a Multiframe Grayscale Word Secondary
       Capture Image Storage
    vtkGDCMImageWriter writer = vtkGDCMImageWriter
      .New();
    writer.SetFileName( "headsq.dcm" );
    writer.SetInput( reader.GetOutput() );
    // cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
    // writer.SetInput( cast.GetOutput() );
    writer.SetFileDimensionality( 3 );
    writer.Write();

    return 0;
    }
}
```

27.78 HelloWorld.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example is ... guess what this is for :)
 */

#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"

#include <iostream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
```

```
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  // Instanciate the reader:
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
    }

  // If we reach here, we know for sure only 1 thing:
  // It is a valid DICOM file (potentially an old ACR-NEMA 1.0/2.0 file)
  // (Maybe, it's NOT a Dicom image -could be a DICOMDIR, a RTSTRUCT, etc-)

  // The output of gdcm::Reader is a gdcm::File
  gdcm::File &file = reader.GetFile();

  // the dataset is the the set of element we are interested in:
  gdcm::DataSet &ds = file.GetDataSet();

  // Contruct a static(*) type for Image Comments :
  gdcm::Attribute<0x0020,0x4000> imagecomments;
  imagecomments.SetValue( "Hello, World !" );

  // Now replace the Image Comments from the dataset with our:
  ds.Replace( imagecomments.GetAsDataElement() );

  // Write the modified DataSet back to disk
  gdcm::Writer writer;
  writer.CheckFileMetaInformationOff(); // Do not
       attempt to reconstruct the file meta to preserve the file
                                        // as close to the original as
       possible.
  writer.SetFileName( outfilename );
  writer.SetFile( file );
  if( !writer.Write() )
    {
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
    }

  return 0;
}

/*
 * (*) static type, means that extra DICOM information VR & VM are computed at
       compilation time.
 * The compiler is deducing those values from the template arguments of the
       class.
 */
```

27.79 HelloWorld.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################

"""
Hello World !
"""
```

```
import gdcm
import sys

if __name__ == "__main__":

  # verbosity:
  #gdcm.Trace.DebugOn()
  #gdcm.Trace.WarningOn()
  #gdcm.Trace.ErrorOn()

  # Get the filename from the command line
  filename = sys.argv[1]

  # Instanciate a gdcm.Reader
  # This is the main class to handle any type of DICOM object
  # You should check for gdcm.ImageReader for reading specifically DICOM Image
      file
  r = gdcm.Reader()
  r.SetFileName( filename )
  # If the reader fails to read the file, we should stop !
  if not r.Read():
    print "Not a valid DICOM file"
    sys.exit(1)

  # Get the DICOM File structure
  file = r.GetFile()

  # Get the DataSet part of the file
  dataset = file.GetDataSet()

  # Ok let's print it !
  print dataset

  # Use StringFilter to print a particular Tag:
  sf = gdcm.StringFilter()
  sf.SetFile(r.GetFile())

  # Check if Attribute exist
  print dataset.FindDataElement( gdcm.Tag(0x0028,0x0010))

  # Let's print it as string pair:
  print sf.ToStringPair(gdcm.Tag(0x0028,0x0010))
```

27.80 iU22tomultisc.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * iU22 Raw Data extractor
 */
#include "gdcmReader.h"
#include "gdcmImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  // IM_001
  const char *filename = argv[1];

  gdcm::Reader reader; // Do not use ImageReader
  reader.SetFileName( filename );
  if( !reader.Read() )
```

```
    {
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
    }

 // * The data is simply 8-bit unsigned in the obvious x/y/z order
 // * 200D,300B contains the data
 // * 200D,3001 contains the no. of voxels (416,412,256 in this case)
 // * 200D,3003 contains the voxel sizes (0.156184527398215 /
 // 0.1223749613981957 / 0.328479990704639 in this case)

  const gdcm::File &file = reader.GetFile();
  const gdcm::DataSet &ds = file.GetDataSet();
  const gdcm::PrivateTag trawdataus( 0x200d, 0x0b, "Philips US
       Imaging DD 033" );
  const gdcm::DataElement &rawdataus = ds.GetDataElement
      ( trawdataus );

  const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x01, "
      Philips US Imaging DD 036" );
  const gdcm::DataElement &colsrowsframes = ds.GetDataElement
      ( tcolsrowsframes );
  // const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x02, "Philips US Imaging
       DD 036" );
  // this is just a duplicate previous tag.
  const gdcm::PrivateTag tvoxelspacing( 0x200d, 0x03, "Philips
       US Imaging DD 036" );
  const gdcm::DataElement &voxelspacing = ds.GetDataElement
      ( tvoxelspacing );

  gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3>
       dims; // Use DS to interpret value stored in LO
  dims.SetFromDataElement( colsrowsframes );

  gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3>
       spacing;
  spacing.SetFromDataElement( voxelspacing );

  gdcm::ImageWriter writer;

  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 3 ); // good default
  image.SetDimension(0, dims[0] );
  image.SetDimension(1, dims[1] );
  image.SetDimension(2, dims[2] );
  image.SetSpacing(0, spacing[0] );
  image.SetSpacing(1, spacing[1] );
  image.SetSpacing(2, spacing[2] );
  gdcm::PixelFormat pixeltype = gdcm::PixelFormat::UINT8
      ;

  gdcm::PhotometricInterpretation pi;
  pi = gdcm::PhotometricInterpretation::MONOCHROME2
      ;
  image.SetPhotometricInterpretation( pi );
  image.SetPixelFormat( pixeltype );

  image.SetDataElement( rawdataus );

  std::string outfilename = "outiu22.dcm";

  gdcm::DataElement de( gdcm::Tag(0x8,0x16) ); // SOP
       Class UID
  de.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms(
    gdcm::MediaStorage::UltrasoundMultiFrameImageStorage
       );
//    gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage
       );
  de.SetByteValue( ms.GetString(), strlen(ms.GetString
      ()));
  writer.GetFile().GetDataSet().Replace( de );

  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr << "could not write: " << outfilename << std::endl;
    return 1;
    }
```

```
  return 0;
}
```

27.81 LargeVRDSExplicit.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFileExplicitFilter.h"
#include "gdcmSequenceOfItems.h"

bool interpolate(const double * pts, unsigned int npts, std::vector<double> &
      out )
{
  out.clear();
  for(unsigned int i = 0; i < 2*npts; ++i )
    {
    const unsigned int j = i / 2;
    if( i % 2 )
      {
      if( j != npts - 1 )
        {
        assert( 3*j+5 < 3*npts );
        const double midpointx = (pts[3*j+0] + pts[3*j+3]) / 2;
        const double midpointy = (pts[3*j+1] + pts[3*j+4]) / 2;
        const double midpointz = (pts[3*j+2] + pts[3*j+5]) / 2;
        out.push_back( midpointx );
        out.push_back( midpointy );
        out.push_back( midpointz );
        }
      }
    else
      {
      assert( j < npts );
      out.push_back( pts[3*j+0] );
      out.push_back( pts[3*j+1] );
      out.push_back( pts[3*j+2] );
      }
    }
  assert( out.size() == 2 * npts * 3 - 3 );
  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  gdcm::FileExplicitFilter fef;
  //fef.SetChangePrivateTags( changeprivatetags );
```

```
fef.SetFile( reader.GetFile() );
if( !fef.Change() )
  {
  std::cerr << "Failed to change: " << filename << std::endl;
  return 1;
  }

// (3006,0039) SQ (Sequence with undefined length #=4)     # u/l, 1
     ROIContourSequence
gdcm::Tag tag(0x3006,0x0039);

const gdcm::DataElement &roicsq = ds.GetDataElement
    ( tag );
gdcm::SmartPointer<gdcm::SequenceOfItems>
     sqi = roicsq.GetValueAsSQ();
//sqi->SetNumberOfItems( 1 );
const gdcm::Item & item = sqi->GetItem(1); // Item start at
     #1
const gdcm::DataSet& nestedds = item.GetNestedDataSet
    ();

gdcm::Tag tcsq(0x3006,0x0040);
if( !nestedds.FindDataElement( tcsq ) )
  {
  return 0;
  }
const gdcm::DataElement& csq = nestedds.GetDataElement
    ( tcsq );
gdcm::SmartPointer<gdcm::SequenceOfItems>
     sqi2 = csq.GetValueAsSQ();
if( !sqi2 || !sqi2->GetNumberOfItems() )
  {
  return 0;
  }
//unsigned int nitems = sqi2->GetNumberOfItems();
gdcm::Item & item2 = sqi2->GetItem(1); // Item start at #1

gdcm::DataSet& nestedds2 = item2.GetNestedDataSet
    ();
//item2.SetVLToUndefined();
//std::cout << nestedds2 << std::endl;
// (3006,0050) DS
     [43.57636\65.52504\-10.0\46.043102\62.564945\-10.0\49.126537\60.714... # 398,48 ContourData
gdcm::Tag tcontourdata(0x3006,0x0050);
const gdcm::DataElement & contourdata = nestedds2.
    GetDataElement( tcontourdata );
//std::cout << contourdata << std::endl;

//const gdcm::ByteValue *bv = contourdata.GetByteValue();
gdcm::Attribute<0x3006,0x0046> ncontourpoints;
ncontourpoints.Set( nestedds2 );

gdcm::Attribute<0x3006,0x0050> at;
at.SetFromDataElement( contourdata );
const double* pts = at.GetValues();
unsigned int npts = at.GetNumberOfValues() / 3;

std::vector<double> out( pts, pts + npts * 3 );
std::vector<double> out2;

//const unsigned int niter = 7;
const unsigned int niter = 8;
for( unsigned int i = 0; i < niter; ++i)
  {
  //bool b =
  interpolate(&out[0], out.size() / 3, out2);
  //const double *pout = &out[0];
  out = out2;
  out2.clear();
  }
assert( out.size() % 3 == 0 );

gdcm::Attribute<0x3006,0x0050> at_interpolate;
at_interpolate.SetNumberOfValues( out.size() / 3 );
at_interpolate.SetValues( &out[0], out.size() );

ncontourpoints.SetValue( at_interpolate.GetNumberOfValues() / 3 );
nestedds2.Replace( at_interpolate.GetAsDataElement() );
nestedds2.Replace( ncontourpoints.GetAsDataElement() )
    ;
```

```
  //assert(0);

  // Let's take item one and subdivide it

  gdcm::TransferSyntax ts =
      gdcm::TransferSyntax::ImplicitVRLittleEndian
      ;
  ts = gdcm::TransferSyntax::ExplicitVRLittleEndian
      ;

  gdcm::FileMetaInformation &fmi = file.GetHeader
      ();
  const char *tsuid = gdcm::TransferSyntax::GetTSString
      ( ts );
  // const char * is ok since padding is \0 anyway...
  gdcm::DataElement de( gdcm::Tag(0x0002,0x0010) );
  de.SetByteValue( tsuid, strlen(tsuid) );
  de.SetVR( gdcm::Attribute<0x0002, 0x0010>::GetVR
      () );
  fmi.Replace( de );
  fmi.Remove( gdcm::Tag(0x0002,0x0012) ); // will be regenerated
  fmi.Remove( gdcm::Tag(0x0002,0x0013) ); //  '   '    '
  fmi.SetDataSetTransferSyntax(ts);


  gdcm::Writer w;
  w.SetFile( file );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.82 MagnifyFile.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageMagnify.h"
#include "vtkImageCast.h"

#include "gdcmTesting.h"
#include "gdcmSystem.h"

// This is a simple test to magnify an image that is known to give excellent
// compression ratio. This will be our test for those large image
int main(int, char *[])
{
  const char *directory = gdcm::Testing::GetDataRoot(
      );
  if(!directory) return 1;
  std::string file = std::string(directory) + "/test.acr";
  std::cout << file << std::endl;
  if( !gdcm::System::FileExists( file.c_str() ) )
      return 1;

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New
      ();
  reader->SetFileName( file.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );
```

```
  vtkImageCast *cast = vtkImageCast::New();
  cast->SetInput( reader->GetOutput() );
  cast->SetOutputScalarTypeToUnsignedShort();

  vtkImageMagnify *magnify = vtkImageMagnify::New();
  magnify->SetInput( cast->GetOutput() );
  magnify->SetInterpolate( 1 );
  magnify->SetInterpolate( 0 );
  int factor = 100;
  magnify->SetMagnificationFactors (factor, factor, 1);

  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
      ();
  writer->SetFileName( "/tmp/bla.dcm" );
  writer->SetInput( magnify->GetOutput() );
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->SetMedicalImageProperties( reader->
      GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines()
       );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->Write();

  // TODO:
  //vtkImageAppendComponents.h

  reader->Delete();
  magnify->Delete();
  writer->Delete();

  return 0;
}
```

27.83 ManipulateFile.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ManipulateFile.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class ManipulateFile
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    string file2 = args[1];
    Reader reader = new Reader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    Anonymizer ano = new Anonymizer();
    ano.SetFile( reader.GetFile() );
    ano.RemovePrivateTags();
    ano.RemoveGroupLength();
```

```
    Tag t = new Tag(0x10,0x10);
    ano.Replace( t, "GDCM^Csharp^Test^Hello^World" );

    UIDGenerator g = new UIDGenerator();
    ano.Replace( new Tag(0x0008,0x0018), g.Generate() );
    ano.Replace( new Tag(0x0020,0x000d), g.Generate() );
    ano.Replace( new Tag(0x0020,0x000e), g.Generate() );
    ano.Replace( new Tag(0x0020,0x0052), g.Generate() );

    Writer writer = new Writer();
    writer.SetFileName( file2 );
    writer.SetFile( ano.GetFile() );
    ret = writer.Write();
    if( !ret )
      {
      return 1;
      }

    return 0;
    }
}
```

27.84 ManipulateFile.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################

"""
Usage:

 python ManipulateFile.py input.dcm output.dcm

Footnote:
  GDCM 1.2.x would create incorrect Multiframes MR Image Storage file. Try to
       recover from
  the issues to recreate a MultiframeGrayscaleByteSecondaryCaptureImageStorage
       file.
  e.g:

  python ManipulateFile.py
       Insight/Testing/Temporary/itkGDCMImageIOTest5-j2k.dcm manipulated.dcm
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  ano = gdcm.Anonymizer()
  ano.SetFile( r.GetFile() )
  ano.RemovePrivateTags()
  ano.Remove( gdcm.Tag(0x0032,0x1030) )
  ano.Remove( gdcm.Tag(0x008,0x14) )
  ano.Remove( gdcm.Tag(0x008,0x1111) )
  ano.Remove( gdcm.Tag(0x008,0x1120) )
  ano.Remove( gdcm.Tag(0x008,0x1140) )
  ano.Remove( gdcm.Tag(0x10,0x21b0) )
  ano.Empty( gdcm.Tag(0x10,0x10) )
```

```
  ano.Empty( gdcm.Tag(0x10,0x20) )
  ano.Empty( gdcm.Tag(0x10,0x30) )
  ano.Empty( gdcm.Tag(0x20,0x10) )
  ano.Empty( gdcm.Tag(0x32,0x1032) )
  ano.Empty( gdcm.Tag(0x32,0x1033) )
  ano.Empty( gdcm.Tag(0x40,0x241) )
  ano.Empty( gdcm.Tag(0x40,0x254) )
  ano.Empty( gdcm.Tag(0x40,0x253) )
  ano.Empty( gdcm.Tag(0x40,0x1001) )
  ano.Empty( gdcm.Tag(0x8,0x80) )
  ano.Empty( gdcm.Tag(0x8,0x50) )
  ano.Empty( gdcm.Tag(0x8,0x1030) )
  ano.Empty( gdcm.Tag(0x8,0x103e) )
  ano.Empty( gdcm.Tag(0x18,0x1030) )
  ano.Empty( gdcm.Tag(0x38,0x300) )
  g = gdcm.UIDGenerator()
  ano.Replace( gdcm.Tag(0x0008,0x0018), g.Generate() )
  ano.Replace( gdcm.Tag(0x0020,0x00d), g.Generate() )
  ano.Replace( gdcm.Tag(0x0020,0x00e), g.Generate() )
  ano.Replace( gdcm.Tag(0x0020,0x052), g.Generate() )
  #ano.Replace( gdcm.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
  """
  ano.Remove( gdcm.Tag(0x0018,0x0020) ) # ScanningSequence
  ano.Remove( gdcm.Tag(0x0018,0x0021) ) # SequenceVariant
  ano.Remove( gdcm.Tag(0x0018,0x0022) ) # ScanOptions
  ano.Remove( gdcm.Tag(0x0018,0x0023) ) # MRAcquisitionType
  ano.Remove( gdcm.Tag(0x0018,0x0050) ) # SliceThickness
  ano.Remove( gdcm.Tag(0x0018,0x0080) ) # RepetitionTime
  ano.Remove( gdcm.Tag(0x0018,0x0081) ) # EchoTime
  ano.Remove( gdcm.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
  ano.Remove( gdcm.Tag(0x0018,0x0091) ) # EchoTrainLength
  ano.Remove( gdcm.Tag(0x0018,0x1164) ) # ImagerPixelSpacing

  ano.Remove( gdcm.Tag(0x0020,0x0032) ) # Image Position (Patient)
  ano.Remove( gdcm.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
  ano.Remove( gdcm.Tag(0x0020,0x0052) ) # Frame of Reference UID
  ano.Remove( gdcm.Tag(0x0020,0x1040) ) # Position Reference Indicator

  ano.Replace( gdcm.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation

  ano.Empty( gdcm.Tag(0x0020,0x0020) )

  ano.Remove( gdcm.Tag(0x7fe0,0x0000) )

  #ano.Empty( gdcm.Tag(0x0028,0x0009) ) # Frame Increment Pointer

  #ano.Empty( gdcm.Tag(0x0028,0x1052) )  #<entry group="0028" element="1052"
       vr="DS" vm="1" name="Rescale Intercept"/>
  #ano.Empty( gdcm.Tag(0x0028,0x1053) )  #<entry group="0028" element="1053"
       vr="DS" vm="1" name="Rescale Slope"/>
  #ano.Replace( gdcm.Tag(0x0028,0x1054), "US" )  #<entry group="0028" element="
      1054" vr="LO" vm="1" name="Rescale Type"/>

  ano.Replace( gdcm.Tag(0x2050, 0x0020), "IDENTITY")
  """

  w = gdcm.Writer()
  w.SetFile( ano.GetFile() )
  w.SetFileName( file2 )
  if not w.Write():
    sys.exit(1)
```

27.85 ManipulateSequence.py

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################
```

```
"""
Usage:

 python ManipulateSequence.py input.dcm output.dcm

This was tested using:

 python ManipulateSequence.py gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

This is a dummy example on how to modify a value set in a nested-nested dataset

WARNING:
Do not use as-is in production, this is just an example
This example works in an undefined length Item only (you need to explicitely
       recompute the length otherwise)
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  f = r.GetFile()
  ds = f.GetDataSet()
  tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
  if ds.FindDataElement( tsis ):
    sis = ds.GetDataElement( tsis )
    #sqsis = sis.GetSequenceOfItems()
    # GetValueAsSQ handle more cases
    sqsis = sis.GetValueAsSQ()
    if sqsis.GetNumberOfItems():
      item1 = sqsis.GetItem(1)
      nestedds = item1.GetNestedDataSet()
      tprcs = gdcm.Tag(0x0040,0xa170) # PurposeOfReferenceCodeSequence
      if nestedds.FindDataElement( tprcs ):
        prcs = nestedds.GetDataElement( tprcs )
        sqprcs = prcs.GetSequenceOfItems()
        if sqprcs.GetNumberOfItems():
          item2 = sqprcs.GetItem(1)
          nestedds2 = item2.GetNestedDataSet()
          # (0008,0104) LO [Uncompressed predecessor]               #  24, 1
       CodeMeaning
          tcm = gdcm.Tag(0x0008,0x0104)
          if nestedds2.FindDataElement( tcm ):
            cm = nestedds2.GetDataElement( tcm )
            mystr = "GDCM was here"
            cm.SetByteValue( mystr, gdcm.VL( len(mystr) ) )

  w = gdcm.Writer()
  w.SetFile( f )
  w.SetFileName( file2 )
  if not w.Write():
    sys.exit(1)
```

27.86 MergeFile.py

```
#########################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
```

```
##############################################################################

"""
Usage:

 python MergeFile.py input1.dcm input2.dcm

  It will produce a 'merge.dcm' output file, which contains all meta
       information from input1.dcm
  and copy the Stored Pixel values from input2.dcm
  This script even works when input2.dcm is a Secondary Capture and does not
       contains information
  such as IOP and IPP...
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r1 = gdcm.ImageReader()
  r1.SetFileName( file1 )
  if not r1.Read():
    sys.exit(1)

  r2 = gdcm.ImageReader()
  r2.SetFileName( file2 )
  if not r2.Read():
    sys.exit(1)

  # Image from r2 could be Secondary Capture and thus would not contains
       neither IPP nor IOP
  # Instead always prefer to only copy the Raw Data Element.
  # Warning ! Image need to be identical ! Only the value of Stored Pixel can
       be different.
  r1.GetImage().SetDataElement( r2.GetImage().GetDataElement() )

  w = gdcm.ImageWriter()
  w.SetFile( r1.GetFile() )
  #w.SetImage( r2.GetImage() )  # See comment above
  w.SetImage( r1.GetImage() )

  w.SetFileName( "merge.dcm" )
  if not w.Write():
    sys.exit(1)

  sys.exit(0)
```

27.87 MergeTwoFiles.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example will show how one can read in two DICOM files, use the dataset
 * from file1 and use image from file2  to save it in a 3rd file.
 *
 * Eg:
 * MergeTwoFiles gdcmData/012345.002.050.dcm gdcmData/test.acr merge.dcm
 */

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
```

```
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    return 1;
    }
  const char *file1 = argv[1];
  const char *file2 = argv[2];
  const char *file3 = argv[3];

  // Read file1
  gdcm::ImageReader reader1;
  reader1.SetFileName( file1 );
  if( !reader1.Read() )
    {
    return 1;
    }

  // Read file2
  gdcm::ImageReader reader2;
  reader2.SetFileName( file2 );
  if( !reader2.Read() )
    {
    return 1;
    }

  // Ok now let's take the DataSet from file1 and the Image from file2
  // Warning: if file2 is -for example- a Secondary Capture Storage, then it
       has no
  // Image Orientation (Patient) thus any Image Orientation (Patient) from
       file1
  // will be discarded...

  // let's be fancy. In case reader2 contains explicit, but reader1 is implicit
  // we would rather see an implicit output
  if( reader1.GetFile().GetHeader().GetDataSetTransferSyntax
      () == gdcm::TransferSyntax::ImplicitVRLittleEndian
      )
    {
    reader2.GetImage().SetTransferSyntax(
      gdcm::TransferSyntax::ImplicitVRLittleEndian
      );
    }

  gdcm::ImageWriter writer;
  writer.SetFileName( file3 );
  writer.SetFile( reader1.GetFile() );
  // ImageWriter will always use all of gdcm::Image information an override
       anything wrong from
  // reader1.GetFile(), including the Transfer Syntax
  writer.SetImage( reader2.GetImage() );

  gdcm::DataSet &ds = reader1.GetFile().GetDataSet
      ();

  // Make sure that SOPInstanceUID are different
  // Simply removing it is sufficient as gdcm::ImageWriter will generate one by
       default
  // if not found.
  ds.Remove( gdcm::Tag(0x0008,0x0018) );
  if( !writer.Write() )
    {
    return 1;
    }

  return 0;
}
```

27.88 MetaImageMD5Activiz.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library
```

```
  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using vtkgdcm;
using gdcm;

/*
 * $ export MONO_PATH=/usr/lib/cli/activiz-cil/
 * $ export LD_LIBRARY_PATH=/usr/lib/cli/activiz-cil/
 * $ mono ./bin/MetaImageMD5Activiz.exe gdcmData/012345.002.050.dcm
 */
public class MetaImageMD5Activiz
{
  public static int ProcessOneMHDMD5(string filename)
    {
    vtkGDCMImageReader reader = vtkGDCMImageReader
      .New();
    reader.FileLowerLeftOn();
    reader.DebugOff();
    int canread = reader.CanReadFile( filename );
    if( canread == 0 )
      {
      string refms = gdcm.Testing.GetMediaStorageFromFile(filename);
      if( gdcm.MediaStorage.IsImage( gdcm.MediaStorage.GetMSType(refms) ) )
        {
        System.Console.Write( "Problem with file: " + filename + "\n" );
        return 1;
        }
      // not an image
      return 0;
      }

    reader.SetFileName( filename );
    reader.Update();

    // System.Console.Write(reader.GetOutput());

    vtkMetaImageWriter writer = vtkMetaImageWriter.New();
    writer.SetCompression( false );
    writer.SetInput( reader.GetOutput() );
    string subdir = "MetaImageMD5Activiz";
    string tmpdir = gdcm.Testing.GetTempDirectory( subdir );
    if( !gdcm.PosixEmulation.FileIsDirectory( tmpdir ) )
      {
      gdcm.PosixEmulation.MakeDirectory( tmpdir );
      }
    string mhdfile = gdcm.Testing.GetTempFilename( filename, subdir );

    string rawfile = mhdfile;
    mhdfile += ".mhd";
    rawfile += ".raw";
    writer.SetFileName( mhdfile );
    writer.Write();

    string digestmhd = gdcm.Testing.ComputeFileMD5( mhdfile );
    string digestraw = gdcm.Testing.ComputeFileMD5( rawfile );

    string mhdref = vtkGDCMTesting.GetMHDMD5FromFile
      (filename);
    string rawref = vtkGDCMTesting.GetRAWMD5FromFile
      (filename);

    if( mhdref != digestmhd )
      {
      System.Console.Write( "Problem with mhd file: " + filename + "\n" );
      System.Console.Write( digestmhd );
      System.Console.Write( "\n" );
      System.Console.Write( mhdref );
      System.Console.Write( "\n" );
      return 1;
      }
    if( rawref != digestraw )
      {
```

```
      System.Console.Write( "Problem with raw file: " + filename + "\n" );
      System.Console.Write( digestraw );
      System.Console.Write( "\n" );
      System.Console.Write( rawref );
      System.Console.Write( "\n" );
      return 1;
      }

    return 0;
    }
  public static int Main(string[] args)
    {
    if ( args.Length  == 1 )
      {
      string filename = args[0];
      return ProcessOneMHDMD5( filename );
      }
    // Loop over all gdcmData
    gdcm.Trace.DebugOff();
    gdcm.Trace.WarningOff();
    gdcm.Trace.ErrorOff();

    uint n = gdcm.Testing.GetNumberOfFileNames();
    int ret = 0;
    for( uint i = 0; i < n; ++i )
      {
      string filename = gdcm.Testing.GetFileName( i );
      ret += ProcessOneMHDMD5( filename );
      }
    return ret;
    }
}
```

27.89 MIPViewer.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import vtk.*;
import gdcm.*;
import java.io.File;
import java.awt.Canvas;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MIPViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/
      jni:. CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MIPViewer BRAINX
 *
 */
public class MIPViewer extends Canvas
{
  static {
    // VTK
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    System.loadLibrary("vtkVolumeRenderingJava"); // vtkSmartVolumeMapper
    System.loadLibrary("vtkWidgetsJava"); // vtkBoxWidget
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
  }
```

```
static FilenamesType fns = new FilenamesType();

protected native int Lock();

protected native int UnLock();

public static void process(String path)
  {
  fns.add( path );
  }

// Process only files under dir
public static void visitAllFiles(File dir)
  {
  if (dir.isDirectory())
    {
    String[] children = dir.list();
    for (int i=0; i<children.length; i++)
      {
      visitAllFiles(new File(dir, children[i]));
      }
    }
  else
    {
    process(dir.getPath());
    }
  }

public static void main(String[] args) throws Exception
  {
  String dirname = args[0];
  if( !PosixEmulation.FileIsDirectory( dirname ) )
    {
    return;
    }

  File dir = new File(dirname);
  visitAllFiles(dir);

  IPPSorter ipp = new IPPSorter();
  ipp.SetComputeZSpacing( true );
  ipp.SetZSpacingTolerance( 1e-3 );
  boolean b = ipp.Sort( fns );
  if(!b)
    {
    throw new Exception("Could not scan");
    }
  double ippzspacing = ipp.GetZSpacing();

  FilenamesType sorted = ipp.GetFilenames();
  vtkStringArray files = new vtkStringArray();
  long nfiles = sorted.size();
  //for( String f : sorted )
  for (int i = 0; i < nfiles; i++) {
    String f = sorted.get(i);
    files.InsertNextValue( f );
    }
  vtkGDCMImageReader reader = new vtkGDCMImageReader
    ();
  reader.SetFileNames( files );
  reader.Update(); // get spacing value

  double[] spacing = reader.GetOutput().GetSpacing();

  vtkImageChangeInformation change = new vtkImageChangeInformation();
  change.SetInputConnection( reader.GetOutputPort() );
  change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

  // Create our volume and mapper
  vtkVolume volume = new vtkVolume();
  vtkSmartVolumeMapper mapper = new vtkSmartVolumeMapper();

  vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();

  // Add a box widget if the clip option was selected
  vtkBoxWidget box = new vtkBoxWidget();
  box.SetInteractor(iren);
  box.SetPlaceFactor(1.01);
  box.SetInput(change.GetOutput());
```

```
    //box.SetDefaultRenderer(renderer);
    box.InsideOutOn();
    box.PlaceWidget();
    //vtkBoxWidgetCallback callback = vtkBoxWidgetCallback::New();
    //callback.SetMapper(mapper);
    //box.AddObserver(vtkCommand::InteractionEvent, callback);
    //callback.Delete();
//    Lock();
//    box.EnabledOn();
//    UnLock();
    box.GetSelectedFaceProperty().SetOpacity(0.0);

    mapper.SetInputConnection( change.GetOutputPort() );

    // Create our transfer function
    vtkColorTransferFunction colorFun = new vtkColorTransferFunction();
    vtkPiecewiseFunction opacityFun = new vtkPiecewiseFunction();

    // Create the property and attach the transfer functions
    vtkVolumeProperty property = new vtkVolumeProperty();
    property.IndependentComponentsOn();
    property.SetColor( colorFun );
    property.SetScalarOpacity( opacityFun );
    property.SetInterpolationTypeToLinear();

    // connect up the volume to the property and the mapper
    volume.SetProperty( property );
    volume.SetMapper( mapper );

    vtkMedicalImageProperties medprop = reader.GetMedicalImageProperties();
    int n = medprop.GetNumberOfWindowLevelPresets();
    double opacityWindow = 4096;
    double opacityLevel = 2048;

    // Override default with value from DICOM files:
    for( int i = 0; i < n; ++i )
      {
      double wl[] = medprop.GetNthWindowLevelPreset(i);
      //System.out.println( "W/L: " + wl[0] + " " + wl[1] );
      opacityWindow = wl[0];
      opacityLevel  = wl[1];
      }

    colorFun.AddRGBSegment(0.0, 1.0, 1.0, 1.0, 255.0, 1.0, 1.0, 1.0 );
    opacityFun.AddSegment( opacityLevel - 0.5*opacityWindow, 0.0,
      opacityLevel + 0.5*opacityWindow, 1.0 );
    mapper.SetBlendModeToMaximumIntensity();

    // Create the RenderWindow, Renderer
    vtkRenderer ren1 = new vtkRenderer();
    vtkRenderWindow renWin = new vtkRenderWindow();
    renWin.AddRenderer(ren1);

    // Set the default window size
    renWin.SetSize(600,600);

    // Add the volume to the scene
    ren1.AddVolume( volume );
    ren1.ResetCamera();

    iren.SetRenderWindow( renWin );

    // interact with data
    renWin.Render();

    iren.Start();
    }
}
```

27.90 MPRViewer.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
```

```
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import vtk.*;
import gdcm.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/
      jni:. CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MPRViewer BRAINX
 *
 */
public class MPRViewer
{
  static {
    // VTK
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
  }

  static FilenamesType fns = new FilenamesType();

  public static void process(String path)
    {
    fns.add( path );
    }

  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }

  public static void main(String[] args) throws Exception
    {
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
      {
      return;
      }

    File dir = new File(dirname);
    visitAllFiles(dir);

    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
      {
      throw new Exception("Could not scan");
      }
    double ippzspacing = ipp.GetZSpacing();

    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
```

```
    long nfiles = sorted.size();
    //for( String f : sorted )
    for (int i = 0; i < nfiles; i++) {
      String f = sorted.get(i);
      files.InsertNextValue( f );
      }
    vtkGDCMImageReader reader = new vtkGDCMImageReader
      ();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value

    double[] spacing = reader.GetOutput().GetSpacing();

    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

    // A simple vtkInteractorStyleImage example for
    // 3D image viewing with the vtkImageResliceMapper.
    //
    // Drag Left mouse button to window/level
    // Shift-Left drag to rotate (oblique slice)
    // Shift-Middle drag to slice through image
    // OR Ctrl-Right drag to slice through image

    // Create the RenderWindow, Renderer
    vtkRenderer ren1 = new vtkRenderer();
    vtkRenderWindow renWin = new vtkRenderWindow();
    renWin.AddRenderer(ren1);

    vtkImageResliceMapper im = new vtkImageResliceMapper();
    im.SetInputConnection(change.GetOutputPort());
    im.SliceFacesCameraOn();
    im.SliceAtFocalPointOn();
    im.BorderOff();

    vtkImageProperty ip = new vtkImageProperty();
    ip.SetColorWindow(2000);
    ip.SetColorLevel(1000);
    ip.SetAmbient(0.0);
    ip.SetDiffuse(1.0);
    ip.SetOpacity(1.0);
    ip.SetInterpolationTypeToLinear();

    vtkImageSlice ia = new vtkImageSlice();
    ia.SetMapper(im);
    ia.SetProperty(ip);

    ren1.AddViewProp(ia);
    ren1.SetBackground(0.1,0.2,0.4);
    renWin.SetSize(300,300);

    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
    vtkInteractorStyleImage style = new vtkInteractorStyleImage();
    style.SetInteractionModeToImage3D();
    iren.SetInteractorStyle(style);
    renWin.SetInteractor(iren);

    // render the image
    renWin.Render();
    vtkCamera cam1 = ren1.GetActiveCamera();
    cam1.ParallelProjectionOn();
    ren1.ResetCameraClippingRange();
    renWin.Render();

    iren.Start();
    }
}
```

27.91 MPRViewer2.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
```

```
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import vtk.*;
import gdcm.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer2.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/
      jni:. CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MPRViewer2
       BRAINX
 *
 */
public class MPRViewer2
{
  static {
    // VTK
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    System.loadLibrary("vtkHybridJava");
    System.loadLibrary("vtkWidgetsJava");
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
  }

  static FilenamesType fns = new FilenamesType();

  public static void process(String path)
    {
    fns.add( path );
    }

  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }

  public void dointer(vtkImagePlaneWidget  current_widget)
    {
    int cstat = current_widget.GetCursorDataStatus();
    double[] v = current_widget.GetCurrentCursorPosition();
    //System.out.println( cstat );
    //System.out.println( v[0] );
    //System.out.println( v[1] );
    //System.out.println( v[2] );
    planeWidgetX.SetSliceIndex( (int)v[0] );
    planeWidgetY.SetSliceIndex( (int)v[1] );
    planeWidgetZ.SetSliceIndex( (int)v[2] );
    planeWidgetX.GetCurrentRenderer().ResetCameraClippingRange();
    planeWidgetY.GetCurrentRenderer().ResetCameraClippingRange();
    planeWidgetZ.GetCurrentRenderer().ResetCameraClippingRange();
    }
  public void startinterX()
    {
    dointer( planeWidgetX );
    }
  public void interX()
```

```
    {
    dointer( planeWidgetX );
    }
  public void endinterX()
    {
    }
  public void startinterY()
    {
    dointer( planeWidgetY );
    }
  public void interY()
    {
    dointer( planeWidgetY );
    }
  public void endinterY()
    {
    }
  public void startinterZ()
    {
    dointer( planeWidgetZ );
    }
  public void interZ()
    {
    dointer( planeWidgetZ );
    }
  public void endinterZ()
    {
    //System.out.println( "endinter" );
    }

  public static void AlignCamera(int slice_number, vtkImagePlaneWidget
      current_widget)
    {
    vtkImageData image = (vtkImageData)current_widget.GetInput();
    vtkRenderer ren = current_widget.GetCurrentRenderer();
    double[] origin = image.GetOrigin();
    double ox = origin[0];
    double oy = origin[1];
    double oz = origin[2];

    int wextent[] = image.GetWholeExtent();
    int xMin = wextent[0];
    int xMax = wextent[1];
    int yMin = wextent[2];
    int yMax = wextent[3];
    int zMin = wextent[4];
    int zMax = wextent[5];

    double[] spacing = image.GetSpacing();
    double sx = spacing[0];
    double sy = spacing[1];
    double sz = spacing[2];

    double cx = ox+(0.5*(xMax-xMin))*sx;
    double cy = oy+(0.5*(yMax-yMin))*sy;
    double cz = oy+(0.5*(zMax-zMin))*sz;
    double vx = 0, vy = 0, vz = 0;
    double nx = 0, ny = 0, nz = 0;
    int iaxis = current_widget.GetPlaneOrientation();
    if ( iaxis == 0 ) {
      vz = -1;
      nx = ox + xMax*sx;
      cx = ox + slice_number*sx;
      }
    else if ( iaxis == 1 ) {
      vz = -1;
      ny = oy+yMax*sy;
      cy = oy+slice_number*sy;
      }
    else {
      vy = 1;
      nz = oz+zMax*sz;
      cz = oz+slice_number*sz;
    }
    double px = cx+nx*2;
    double py = cy+ny*2;
    double pz = cz+nz*3;

    vtkCamera camera = ren.GetActiveCamera();
    camera.SetViewUp(vx, vy, vz);
    camera.SetFocalPoint(cx, cy, cz);
```

```
  camera.SetPosition(px, py, pz);
  camera.OrthogonalizeViewUp();
  ren.ResetCameraClippingRange();
  }

private vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
private vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
private vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();

public void config()
  {
  //System.out.println( "config" );
  planeWidgetX.GetCurrentRenderer().ResetCamera();
  planeWidgetY.GetCurrentRenderer().ResetCamera();
  planeWidgetZ.GetCurrentRenderer().ResetCamera();
  }

public void Run(String dirname)
  {
  File dir = new File(dirname);
  visitAllFiles(dir);

  IPPSorter ipp = new IPPSorter();
  ipp.SetComputeZSpacing( true );
  ipp.SetZSpacingTolerance( 1e-3 );
  boolean b = ipp.Sort( fns );
  if(!b)
    {
    //throw new Exception("Could not scan");
    }
  double ippzspacing = ipp.GetZSpacing();

  FilenamesType sorted = ipp.GetFilenames();
  vtkStringArray files = new vtkStringArray();
  long nfiles = sorted.size();
  //for( String f : sorted )
  for (int i = 0; i < nfiles; i++) {
    String f = sorted.get(i);
    files.InsertNextValue( f );
    }
  vtkGDCMImageReader reader = new vtkGDCMImageReader
    ();
  reader.SetFileNames( files );
  reader.Update(); // get spacing value

  double[] spacing = reader.GetOutput().GetSpacing();

  vtkImageChangeInformation change = new vtkImageChangeInformation();
  change.SetInputConnection( reader.GetOutputPort() );
  change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  change.Update();

  System.out.println( change.GetOutput().toString() );

  vtkRenderer ren1 = new vtkRenderer();
  ren1.SetViewport(0., 0., 0.333, 1);
  ren1.SetBackground(0.1,0.2,0.4);
  vtkRenderer ren2 = new vtkRenderer();
  ren2.SetViewport(0.333, 0., 0.667, 1);
  ren2.SetBackground(0.1,0.2,0.4);
  vtkRenderer ren3 = new vtkRenderer();
  ren3.SetViewport(0.667, 0., 1., 1.);
  ren3.SetBackground(0.1,0.2,0.4);

  vtkRenderWindow renWin = new vtkRenderWindow();
  renWin.AddRenderer(ren1);
  renWin.AddRenderer(ren2);
  renWin.AddRenderer(ren3);

  vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
  iren.SetRenderWindow(renWin);

  vtkInteractorStyleImage style = new vtkInteractorStyleImage();
  iren.SetInteractorStyle( style );

  vtkCellPicker picker = new vtkCellPicker();
  picker.SetTolerance(0.005);

  vtkProperty ipwProp = new vtkProperty();

  //vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
```

```
    planeWidgetX.SetInteractor(iren);
    planeWidgetX.SetCurrentRenderer(ren1);
    planeWidgetX.SetDefaultRenderer(ren1);
    planeWidgetX.RestrictPlaneToVolumeOn();
    planeWidgetX.SetTexturePlaneProperty(ipwProp);
    //planeWidgetX.GetPlaneProperty().SetColor(1,0,0);
    //planeWidgetX.TextureInterpolateOff();
    //planeWidgetX.SetResliceInterpolateToNearestNeighbour();
    planeWidgetX.SetInput(change.GetOutput());
    planeWidgetX.SetPlaneOrientationToXAxes();
    planeWidgetX.SetSliceIndex(62);
    planeWidgetX.SetPicker(picker);
    planeWidgetX.SetKeyPressActivationValue('x');
    planeWidgetX.On();
    planeWidgetX.InteractionOn();

    //vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
    planeWidgetY.SetInteractor(iren);
    planeWidgetY.SetCurrentRenderer(ren2);
    planeWidgetY.SetDefaultRenderer(ren2);
    planeWidgetY.RestrictPlaneToVolumeOn();
    planeWidgetY.SetTexturePlaneProperty(ipwProp);
    //planeWidgetY.GetPlaneProperty().SetColor(1,0,0);
    //planeWidgetY.TextureInterpolateOff();
    //planeWidgetY.SetResliceInterpolateToNearestNeighbour();
    planeWidgetY.SetInput(change.GetOutput());
    planeWidgetY.SetLookupTable( planeWidgetX.GetLookupTable() );
    planeWidgetY.SetPlaneOrientationToYAxes();
    planeWidgetY.SetSliceIndex(32);
    planeWidgetY.SetPicker(picker);
    planeWidgetY.SetKeyPressActivationValue('y');
    planeWidgetY.On();


    //vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();
    planeWidgetZ.SetInteractor(iren);
    planeWidgetZ.SetCurrentRenderer(ren3);
    planeWidgetZ.SetDefaultRenderer(ren3);
    planeWidgetZ.RestrictPlaneToVolumeOn();
    planeWidgetZ.SetTexturePlaneProperty(ipwProp);
    //planeWidgetZ.GetPlaneProperty().SetColor(1,0,0);
    //planeWidgetZ.TextureInterpolateOff();
    //planeWidgetZ.SetResliceInterpolateToNearestNeighbour();
    planeWidgetZ.SetInput(change.GetOutput());
    planeWidgetZ.SetLookupTable( planeWidgetX.GetLookupTable() );
    planeWidgetZ.SetPlaneOrientationToZAxes();
    planeWidgetZ.SetSliceIndex(32);
    planeWidgetZ.SetPicker(picker);
    planeWidgetZ.SetKeyPressActivationValue('z');
    planeWidgetZ.On();

    iren.Initialize();

    renWin.Render();
    AlignCamera(52, planeWidgetX);
    AlignCamera(32, planeWidgetY);
    AlignCamera(32, planeWidgetZ);

    planeWidgetX.GetCurrentRenderer().ResetCamera();
    planeWidgetY.GetCurrentRenderer().ResetCamera();
    planeWidgetZ.GetCurrentRenderer().ResetCamera();

    renWin.Render();

    planeWidgetX.AddObserver("StartInteractionEvent", this,"startinterX");
    planeWidgetX.AddObserver("InteractionEvent", this,"interX");
    planeWidgetX.AddObserver("EndInteractionEvent", this,"endinterX");
    planeWidgetY.AddObserver("StartInteractionEvent", this,"startinterY");
    planeWidgetY.AddObserver("InteractionEvent", this,"interY");
    planeWidgetY.AddObserver("EndInteractionEvent", this,"endinterY");
    planeWidgetZ.AddObserver("StartInteractionEvent", this,"startinterZ");
    planeWidgetZ.AddObserver("InteractionEvent", this,"interZ");
    planeWidgetZ.AddObserver("EndInteractionEvent", this,"endinterZ");

    iren.AddObserver("ConfigureEvent", this,"config");

    iren.Start();
    }

  public static void main(String[] args) throws Exception
    {
```

```
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
      {
      return;
      }

    MPRViewer2 me = new MPRViewer2();
    me.Run( dirname );
    }
}
```

27.92 MrProtocol.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 *
 */

/*
28 - 'MrProtocol' VM 1, VR UN, SyngoDT 0, NoOfItems 6, Data '### ASCCONV BEGIN
       ###
ulVersion                                = 0xbee332
tSequenceFileName                        = "%SiemensSeq%\fl_fq_shphs"
tProtocolName                            =
       "flash+AF8-100+AF8-through-plane+AF8-V"
tReferenceImage0                         =
       "1.3.12.2.1107.5.2.9.16041.3000000706210610018120000 4658"
tReferenceImage1                         =
       "1.3.12.2.1107.5.2.9.16041.3000000706210610018120000 4635"
tReferenceImage2                         =
       "1.3.12.2.1107.5.2.9.16041.3000000706210610018120000 4683"
ucScanRegionPosValid                     = 0x1
sProtConsistencyInfo.tBaselineString     = "N4_VB11A_LATEST_20031004"
sProtConsistencyInfo.flNominalB0         = 1.494
sProtConsistencyInfo.flGMax              = 22
sProtConsistencyInfo.flRiseTime          = 10
sGRADSPEC.sEddyCompensationX.aflAmplitude[0] = 0.0141111
sGRADSPEC.sEddyCompensationX.aflAmplitude[1] = 0.057038
sGRADSPEC.sEddyCompensationX.aflAmplitude[2] = -0.00986504
sGRADSPEC.sEddyCompensationX.aflAmplitude[3] = 0.00247627
sGRADSPEC.sEddyCompensationX.aflAmplitude[4] = 0.0026377
sGRADSPEC.sEddyCompensationX.aflTimeConstant[0] = 1.53826
sGRADSPEC.sEddyCompensationX.aflTimeConstant[1] = 0.746617
sGRADSPEC.sEddyCompensationX.aflTimeConstant[2] = 0.339236
sGRADSPEC.sEddyCompensationX.aflTimeConstant[3] = 0.0309809
sGRADSPEC.sEddyCompensationX.aflTimeConstant[4] = 0.00067694
sGRADSPEC.sEddyCompensationY.aflAmplitude[0] = 0.0156411
sGRADSPEC.sEddyCompensationY.aflAmplitude[1] = 0.0440623
sGRADSPEC.sEddyCompensationY.aflAmplitude[2] = -0.00782663
sGRADSPEC.sEddyCompensationY.aflAmplitude[3] = 0.00186828
sGRADSPEC.sEddyCompensationY.aflAmplitude[4] = 0.00154504
sGRADSPEC.sEddyCompensationY.aflTimeConstant[0] = 1.47145
sGRADSPEC.sEddyCompensationY.aflTimeConstant[1] = 0.750538
sGRADSPEC.sEddyCompensationY.aflTimeConstant[2] = 0.339397
sGRADSPEC.sEddyCompensationY.aflTimeConstant[3] = 0.0312962
sGRADSPEC.sEddyCompensationY.aflTimeConstant[4] = 0.000895133
sGRADSPEC.sEddyCompensationZ.aflAmplitude[0] = 0.00618504
sGRADSPEC.sEddyCompensationZ.aflAmplitude[1] = 0.00313121
sGRADSPEC.sEddyCompensationZ.aflAmplitude[2] = 0.000289346
sGRADSPEC.sEddyCompensationZ.aflAmplitude[3] = -0.00019677
sGRADSPEC.sEddyCompensationZ.aflAmplitude[4] = 7.66445e-005
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[0] = 3.37462
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[1] = 0.999351
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[2] = 0.0174646
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[3] = 0.0110094
```

```
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[4] = 0.00199922
sGRADSPEC.bEddyCompensationValid         = 1
sGRADSPEC.sB0CompensationX.aflAmplitude[0] = 0.307474
sGRADSPEC.sB0CompensationX.aflAmplitude[1] = 0.029337
sGRADSPEC.sB0CompensationX.aflAmplitude[2] = -0.187118
sGRADSPEC.sB0CompensationX.aflTimeConstant[0] = 0.98583
sGRADSPEC.sB0CompensationX.aflTimeConstant[1] = 0.0308443
sGRADSPEC.sB0CompensationX.aflTimeConstant[2] = 0.000466792
sGRADSPEC.sB0CompensationY.aflAmplitude[0] = 0.365257
sGRADSPEC.sB0CompensationY.aflAmplitude[1] = -0.318647
sGRADSPEC.sB0CompensationY.aflAmplitude[2] = -0.0118978
sGRADSPEC.sB0CompensationY.aflTimeConstant[0] = 0.61535
sGRADSPEC.sB0CompensationY.aflTimeConstant[1] = 0.488831
sGRADSPEC.sB0CompensationY.aflTimeConstant[2] = 0.00199991
sGRADSPEC.sB0CompensationZ.aflAmplitude[0] = -0.44647
sGRADSPEC.sB0CompensationZ.aflAmplitude[1] = -0.0455154
sGRADSPEC.sB0CompensationZ.aflAmplitude[2] = -0.0304901
sGRADSPEC.sB0CompensationZ.aflTimeConstant[0] = 0.959231
sGRADSPEC.sB0CompensationZ.aflTimeConstant[1] = 0.0720189
sGRADSPEC.sB0CompensationZ.aflTimeConstant[2] = 0.00190141
sGRADSPEC.bB0CompensationValid           = 1
sGRADSPEC.sCrossTermCompensationXY.aflAmplitude[0] = 0.00105046
sGRADSPEC.sCrossTermCompensationXY.aflTimeConstant[0] = 0.842014
sGRADSPEC.sCrossTermCompensationXZ.aflAmplitude[0] = -0.00150189
sGRADSPEC.sCrossTermCompensationXZ.aflTimeConstant[0] = 0.736169
sGRADSPEC.sCrossTermCompensationYX.aflAmplitude[0] = -5.5278e-005
sGRADSPEC.sCrossTermCompensationYX.aflTimeConstant[0] = 0.228697
sGRADSPEC.sCrossTermCompensationYZ.aflAmplitude[0] = 0.000307999
sGRADSPEC.sCrossTermCompensationYZ.aflTimeConstant[0] = 1.19431
sGRADSPEC.sCrossTermCompensationZX.aflAmplitude[0] = -0.000286868
sGRADSPEC.sCrossTermCompensationZX.aflTimeConstant[0] = 0.665979
sGRADSPEC.sCrossTermCompensationZY.aflAmplitude[0] = 0.000355175
sGRADSPEC.sCrossTermCompensationZY.aflTimeConstant[0] = 0.844189
sGRADSPEC.bCrossTermCompensationValid    = 1
sGRADSPEC.lOffsetX                       = 25
sGRADSPEC.lOffsetY                       = 84
sGRADSPEC.lOffsetZ                       = 47
sGRADSPEC.bOffsetValid                   = 1
sGRADSPEC.lDelayX                        = 12
sGRADSPEC.lDelayY                        = 11
sGRADSPEC.lDelayZ                        = 9
sGRADSPEC.bDelayValid                    = 1
sGRADSPEC.flSensitivityX                 = 0.000264087
sGRADSPEC.flSensitivityY                 = 0.000272009
sGRADSPEC.flSensitivityZ                 = 0.000272677
sGRADSPEC.bSensitivityValid              = 1
sGRADSPEC.alShimCurrent[0]               = 183
sGRADSPEC.alShimCurrent[1]               = -25
sGRADSPEC.alShimCurrent[2]               = -85
sGRADSPEC.alShimCurrent[3]               = 378
sGRADSPEC.alShimCurrent[4]               = 82
sGRADSPEC.bShimCurrentValid              = 1
sGRADSPEC.ucMode                         = 0x2
sTXSPEC.asNucleusInfo[0].tNucleus        = "1H"
sTXSPEC.asNucleusInfo[0].lFrequency      = 63684693
sTXSPEC.asNucleusInfo[0].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[0].flReferenceAmplitude = 359.734
sTXSPEC.asNucleusInfo[0].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[0].flAmplitudeCorrection = 1
sTXSPEC.asNucleusInfo[0].bAmplitudeCorrectionValid = 1
sTXSPEC.asNucleusInfo[1].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[1].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[1].bAmplitudeCorrectionValid = 1
sTXSPEC.aRFPULSE[0].tName                = "03GreFCE"
sTXSPEC.aRFPULSE[0].bAmplitudeValid      = 0x1
sTXSPEC.aRFPULSE[0].flAmplitude          = 147.095
sTXSPEC.aRFPULSE[1].tName                = "02GreFCE"
sTXSPEC.aRFPULSE[1].bAmplitudeValid      = 0x1
sTXSPEC.aRFPULSE[1].flAmplitude          = 147.095
sTXSPEC.aRFPULSE[2].tName                = "01GreFCE"
sTXSPEC.aRFPULSE[2].bAmplitudeValid      = 0x1
sTXSPEC.aRFPULSE[2].flAmplitude          = 147.095
sTXSPEC.lNoOfTraPulses                   = 3
sTXSPEC.lBTB1ParallelCapacity            = 2
sTXSPEC.lBTB1SerialCapacity              = 24
sTXSPEC.lBTB2ParallelCapacity            = 2
sTXSPEC.lBTB2SerialCapacity              = 26
sTXSPEC.bBTBValid                        = 1
sTXSPEC.flKDynMagnitudeMin               = 0.5
sTXSPEC.flKDynMagnitudeMax               = 1.5
sTXSPEC.flKDynMagnitudeClipLow           = 0.96
```

```
sTXSPEC.flKDynMagnitudeClipHigh          = 1.04
sTXSPEC.flKDynPhaseMax                   = 0.698132
sTXSPEC.flKDynPhaseClip                  = 0.174533
sTXSPEC.bKDynValid                       = 1
sTXSPEC.ucRFPulseType                    = 0x1
sTXSPEC.ucExcitMode                      = 0x1
sTXSPEC.ucSimultaneousExcitation         = 0x1
sRXSPEC.lGain                            = 1
sRXSPEC.bGainValid                       = 1
sRXSPEC.aFFT_SCALE[0].lRxChannel         = 1
sRXSPEC.aFFT_SCALE[0].flFactor           = 1.06857
sRXSPEC.aFFT_SCALE[0].bValid             = 1
sRXSPEC.aFFT_SCALE[1].lRxChannel         = 2
sRXSPEC.aFFT_SCALE[1].flFactor           = 1.07454
sRXSPEC.aFFT_SCALE[1].bValid             = 1
sRXSPEC.aFFT_SCALE[2].lRxChannel         = 3
sRXSPEC.aFFT_SCALE[2].flFactor           = 1.06622
sRXSPEC.aFFT_SCALE[2].bValid             = 1
sRXSPEC.aFFT_SCALE[3].lRxChannel         = 4
sRXSPEC.aFFT_SCALE[3].flFactor           = 1.06524
sRXSPEC.aFFT_SCALE[3].bValid             = 1
sRXSPEC.aFFT_SCALE[4].lRxChannel         = 5
sRXSPEC.aFFT_SCALE[4].flFactor           = 0.982692
sRXSPEC.aFFT_SCALE[4].bValid             = 1
sRXSPEC.aFFT_SCALE[5].lRxChannel         = 6
sRXSPEC.aFFT_SCALE[5].flFactor           = 0.988603
sRXSPEC.aFFT_SCALE[5].bValid             = 1
sRXSPEC.aFFT_SCALE[6].lRxChannel         = 7
sRXSPEC.aFFT_SCALE[6].flFactor           = 0.981538
sRXSPEC.aFFT_SCALE[6].bValid             = 1
sRXSPEC.aFFT_SCALE[7].lRxChannel         = 8
sRXSPEC.aFFT_SCALE[7].flFactor           = 1.00856
sRXSPEC.aFFT_SCALE[7].bValid             = 1
sRXSPEC.bVariCapVoltagesValid            = 1
sRXSPEC.alDwellTime[0]                   = 8500
sAdjFreSpec.ulMode                       = 0x1
sAdjFreSpec.ucAdjWithBC                  = 0x1
sAdjTraSpec.ucAdjWithBC                  = 0x1
sAdjShimSpec.ulMode                      = 0x1
sAdjShimSpec.ucAdjWithBC                 = 0x1
sAdjWatSupSpec.ulMode                    = 0x1
sAdjWatSupSpec.ucAdjWithBC               = 0x1
alTR[0]                                  = 37000
lContrasts                               = 1
alTE[0]                                  = 4000
acFlowComp[0]                            = 1
lCombinedEchoes                          = 1
sSliceArray.asSlice[0].sPosition.dSag    = 35.31199581
sSliceArray.asSlice[0].sPosition.dCor    = -8.387765754
sSliceArray.asSlice[0].sPosition.dTra    = -23.13178296
sSliceArray.asSlice[0].sNormal.dSag      = 0.771051253
sSliceArray.asSlice[0].sNormal.dCor      = 0.5863890019
sSliceArray.asSlice[0].sNormal.dTra      = -0.2482496801
sSliceArray.asSlice[0].dThickness        = 6
sSliceArray.asSlice[0].dPhaseFOV         = 187.5
sSliceArray.asSlice[0].dReadoutFOV       = 250
sSliceArray.lSize                        = 1
sSliceArray.lSag                         = 1
sSliceArray.lConc                        = 1
sSliceArray.ucMode                       = 0x1
sSliceArray.sTSat.dThickness             = 40
sSliceArray.sTSat.dGap                   = 10
sGroupArray.asGroup[0].nSize             = 1
sGroupArray.asGroup[0].dDistFact         = 0.2
sGroupArray.anMember[1]                  = -1
sGroupArray.lSize                        = 1
sGroupArray.sPSat.dThickness             = 50
sGroupArray.sPSat.dGap                   = 10
sAutoAlign.dAAMatrix[0]                  = 1
sAutoAlign.dAAMatrix[5]                  = 1
sAutoAlign.dAAMatrix[10]                 = 1
sAutoAlign.dAAMatrix[15]                 = 1
sNavigatorPara.ucRespComp                = 0x4
sPrepPulses.ucFatSat                     = 0x4
sPrepPulses.ucWaterSat                   = 0x4
sPrepPulses.ucInversion                  = 0x4
sPrepPulses.ucSatRecovery                = 0x1
sPrepPulses.ucFatSatMode                 = 0x2
sKSpace.lBaseResolution                  = 256
sKSpace.lPhaseEncodingLines              = 192
sKSpace.dPhaseResolution                 = 1
```

```
sKSpace.lPartitions                      = 32
sKSpace.lImagesPerSlab                   = 32
sKSpace.dSliceResolution                 = 1
sKSpace.ucPhasePartialFourier            = 0x10
sKSpace.ucSlicePartialFourier            = 0x10
sKSpace.ucAveragingMode                  = 0x2
sKSpace.ucMultiSliceMode                 = 0x1
sKSpace.ucDimension                      = 0x2
sKSpace.ucAsymmetricEchoAllowed          = 0x1
sKSpace.unReordering                     = 0x1
sFastImaging.lEPIFactor                  = 1
sFastImaging.lTurboFactor                = 1
sFastImaging.lSegments                   = 3
sFastImaging.ulEnableRFSpoiling          = 0x1
sPhysioImaging.lSignal1                  = 2
sPhysioImaging.lMethod1                  = 2
sPhysioImaging.lSignal2                  = 1
sPhysioImaging.lMethod2                  = 1
sPhysioImaging.lPhases                   = 21
sPhysioImaging.lRetroGatedImages         = 16
sPhysioImaging.sPhysioECG.lScanWindow    = 805
sPhysioImaging.sPhysioECG.lTriggerPulses = 1
sPhysioImaging.sPhysioECG.lTriggerWindow = 5
sPhysioImaging.sPhysioECG.lArrhythmiaDetection = 1
sPhysioImaging.sPhysioECG.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioECG.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioPulse.lTriggerPulses = 1
sPhysioImaging.sPhysioPulse.lTriggerWindow = 5
sPhysioImaging.sPhysioPulse.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioPulse.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioExt.lTriggerPulses = 1
sPhysioImaging.sPhysioExt.lTriggerWindow = 5
sPhysioImaging.sPhysioExt.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioExt.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioResp.lRespGateThreshold = 20
sPhysioImaging.sPhysioResp.lRespGatePhase = 2
sPhysioImaging.sPhysioResp.dGatingRatio  = 0.3
sSpecPara.lPhaseCyclingType              = 1
sSpecPara.lPhaseEncodingType             = 1
sSpecPara.lRFExcitationBandwidth         = 1
sSpecPara.ucRemoveOversampling           = 0x1
sSpecPara.lDecouplingType                = 1
sSpecPara.lNOEType                       = 1
sSpecPara.lExcitationType                = 1
sSpecPara.lSpectralSuppression           = 1
sDiffusion.ulMode                        = 0x1
sAngio.sFlowArray.asElm[0].nVelocity     = 100
sAngio.sFlowArray.asElm[0].nDir          = 0x4
sAngio.sFlowArray.lSize                  = 1
sAngio.ucPCFlowMode                      = 0x2
sAngio.ucTOFInflow                       = 0x4
sAngio.ucRephasedImage                   = 0x1
sAngio.ucPhaseImage                      = 0x1
sEllipticalFilter.ucMode                 = 0x1
sPat.lAccelFactPE                        = 1
sPat.lAccelFact3D                        = 1
sPat.ucPATMode                           = 0x1
sPat.ucRefScanMode                       = 0x1
ucAutoMovie                              = 0x1
ucDisableChangeStoreImages               = 0x1
ucReconstructionMode                     = 0x1
ucPHAPSMode                              = 0x1
ucDixon                                  = 0x1
lAverages                                = 2
adFlipAngleDegree[0]                     = 30
lScanTimeSec                             = 103
lTotalScanTimeSec                        = 112
dRefSNR                                  = 165404.1473
dRefSNR_VOI                              = 165404.1473
tdefaultEVAProt                          =
       "%SiemensEvaDefProt%\Inline\Inline.evp"
tcurrentEVAProt                          = "%CURRENTEVAPROT%\EVA2A5.tmp"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tElement = "PP6"
sCOIL_SELECT_MEAS.asList[0].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[0].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tElement = "PP5"
sCOIL_SELECT_MEAS.asList[1].lElementSelected = 1
```

```
sCOIL_SELECT_MEAS.asList[1].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tElement = "PP3"
sCOIL_SELECT_MEAS.asList[2].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[2].lRxChannelConnected = 2
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tElement = "PP4"
sCOIL_SELECT_MEAS.asList[3].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[3].lRxChannelConnected = 3
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tElement = "PP2"
sCOIL_SELECT_MEAS.asList[4].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[4].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tElement = "PP1"
sCOIL_SELECT_MEAS.asList[5].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[5].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tElement = "PA6"
sCOIL_SELECT_MEAS.asList[6].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[6].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tElement = "PA5"
sCOIL_SELECT_MEAS.asList[7].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[7].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tElement = "PA3"
sCOIL_SELECT_MEAS.asList[8].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[8].lRxChannelConnected = 6
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tElement = "PA4"
sCOIL_SELECT_MEAS.asList[9].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[9].lRxChannelConnected = 7
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tElement = "PA2"
sCOIL_SELECT_MEAS.asList[10].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[10].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tElement = "PA1"
sCOIL_SELECT_MEAS.asList[11].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[11].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[0] = 0xff
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[1] = 0x76
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[2] = 0x78
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[3] = 0x87
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[4] = 0x67
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[0] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[1] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[2] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[3] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[4] = 0x2
sEFISPEC.bEFIDataValid                    = 1
### ASCCONV END ###
'
*/

/*
 * Table of equivalence:
 *
ulVersion                                 = 0xbee332
<=>
27 - 'MrProtocolVersion' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '12510002'
*/

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
```

```
#include <map>

#include <math.h>

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
    }

  gdcm::CSAHeader csa;
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet
      ();

  //const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
  const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag
      ();

  if( ds.FindDataElement( t2 ) )
    {
    csa.LoadFromDataElement( ds.GetDataElement
      ( t2 ) );
    //csa.Print( std::cout );
    }

  if( !csa.FindCSAElementByName( "MrProtocol" ) )
    {
    return 1;
    }
  const gdcm::CSAElement &csael = csa.GetCSAElementByName
      ( "MrProtocol" );
  //std::cout << csael << std::endl;

  const gdcm::ByteValue *bv = csael.GetByteValue();
  if( !bv )
    {
    return 1;
    }
  std::string str(bv->GetPointer(), bv->GetLength());
  std::istringstream is(str);
  std::string s;
  typedef std::map< std::string, std::string > MyMapType;
  MyMapType mymap;
  while( std::getline(is, s ) )
    {
    std::string::size_type pos = s.find( '=' );
    if( pos != std::string::npos )
      {
      std::string sub1 = s.substr(0, pos);
      sub1.erase( sub1.find_last_not_of(' ') + 1);
      std::string sub2 = s.substr(pos+1); // skip the '=' char
      sub2.erase( 0, sub2.find_first_not_of(' '));
      //std::cout << sub1 << std::endl;
      mymap.insert( MyMapType::value_type(sub1, sub2) );
      }
    else
      {
      // ### ASCCONV BEGIN ###
      // ### ASCCONV END ###
      }
    }
  const char fourierstr[] = "sKSpace.ucSlicePartialFourier";
  const gdcm::CSAHeaderDict &csadict =
      gdcm::Global::GetInstance().GetDicts().
      GetCSAHeaderDict();
  const gdcm::CSAHeaderDictEntry &fourier = csadict.
      GetCSAHeaderDictEntry( fourierstr );
  std::cout << fourier << std::endl;
  MyMapType::const_iterator it = mymap.find ( fourierstr );
  if( it == mymap.end() ) return 1;
  //std::cout << it->second << std::endl;
  const std::string &partial_fourier = it->second;
  if( partial_fourier == "0x1" )
    {
```

```
    std::cout << "partial fourier is 4/8" << std::endl;
    }
  else if( partial_fourier == "0x2" )
    {
    std::cout << "partial fourier is 5/8" << std::endl;
    }
  else if( partial_fourier == "0x4" )
    {
    std::cout << "partial fourier is 6/8" << std::endl;
    }
  else if( partial_fourier == "0x8" )
    {
    std::cout << "partial fourier is 7/8" << std::endl;
    }
  else if( partial_fourier == "0x10" )
    {
    std::cout << "partial fourier is 8/8" << std::endl;
    }
  else
    {
    std::cerr << "Impossible: " << partial_fourier << std::endl;
    return 1;
    }
/*
This is the Flip Angle:
adFlipAngleDegree[0]                      = 30

One can find it also in the protocol:

...
      <ParamFunctor.""T1mapFunctor"">
      {
        <Class> ""T1mapFunctor@IceImagePostProcFunctors""

        <ParamBool.""EXECUTE"">  { }
        <ParamDouble.""Flip1_deg"">  { <Precision> 16  14.7378520000000000  }
...

*/
  // Below is an attemp to play with the CSAHeader dict:
#if 0
  const char gspec[] = "sGRADSPEC.flSensitivityX";
  it = mymap.find( gspec );
  if( it == mymap.end() ) return 1;
  const std::string &dummy = it->second;
  std::cout << dummy << std::endl;

  const gdcm::CSAHeaderDictEntry &csaentry = csadict.
      GetCSAHeaderDictEntry( gspec );
  std::cout << csaentry  << std::endl;
#endif


/*
    sSliceArray.ucMode -- should be in (1, 2, 4)
    enum SeriesMode
    {
      ASCENDING   = 0x01,
      DESCENDING  = 0x02,
      INTERLEAVED = 0x04
    };
*/
  const char sliceorderstr[] = "sSliceArray.ucMode";
  const gdcm::CSAHeaderDictEntry &sliceorder = csadict.
      GetCSAHeaderDictEntry( sliceorderstr );
  std::cout << sliceorder << std::endl;

  it = mymap.find ( sliceorderstr );
  if( it == mymap.end() ) return 1;
  const std::string &slice_order = it->second;
  if( slice_order == "0x1" )
    {
    std::cout << "slice_order: ASCENDING" << std::endl;
    }
  else if( slice_order == "0x2" )
    {
    std::cout << "slice_order: DESCENDING" << std::endl;
    }
  else if( slice_order == "0x4" )
    {
    std::cout << "slice_order: INTERLEAVED" << std::endl;
```

```
    }
  else
    {
    std::cerr << "Impossible: " << slice_order << std::endl;
    return 1;
    }

  return 0;
}
```

27.93 NewSequence.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/NewSequence.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
//using gdcm;

public class NewSequence
{
  public static byte[] StrToByteArray(string str)
    {
    System.Text.ASCIIEncoding  encoding=new System.Text.ASCIIEncoding();
    return encoding.GetBytes(str);
    }

  public static int Main(string[] argv)
    {
    string file1 = argv[0];
    string file2 = argv[1];

    gdcm.Reader r = new gdcm.Reader();
    r.SetFileName( file1 );
    if ( ! r.Read() )
      {
      return 1;
      }

    gdcm.File f = r.GetFile();
    gdcm.DataSet ds = f.GetDataSet();
    // tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence

    // Create a dataelement
    gdcm.DataElement de = new gdcm.DataElement(new gdcm.Tag(0x0010, 0x2180));
    string occ = "Occupation";
    de.SetByteValue( StrToByteArray(occ), new gdcm.VL((uint)occ.Length));
    de.SetVR(new gdcm.VR(gdcm.VR.VRType.SH));

    // Create an item
    gdcm.Item it = new gdcm.Item();
    it.SetVLToUndefined();      // Needed to not popup error message
    //it.InsertDataElement(de)
    gdcm.DataSet nds = it.GetNestedDataSet();
    nds.Insert(de);

    // Create a Sequence
    gdcm.SmartPtrSQ sq = gdcm.SequenceOfItems.New();
    sq.SetLengthToUndefined();
    sq.AddItem(it);

    // Insert sequence into data set
```

```
    gdcm.DataElement des = new gdcm.DataElement(new gdcm.Tag(0x0400,0x0550));
    des.SetVR(new gdcm.VR(gdcm.VR.VRType.SQ));
    des.SetValue(sq.__ref__());
    des.SetVLToUndefined();

    ds.Insert(des);

    gdcm.Writer w = new gdcm.Writer();
    w.SetFile( f );
    w.SetFileName( file2 );
    if ( !w.Write() )
      return 1;

    return 0;
    }
}
```

27.94 NewSequence.py

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

"""
Usage:

  python NewSequence.py input.dcm output.dcm


Thanks to Robert Irie for code
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  f = r.GetFile()
  ds = f.GetDataSet()
  #tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence

  # Create a dataelement
  de = gdcm.DataElement(gdcm.Tag(0x0010, 0x2180))
  de.SetByteValue("Occupation", gdcm.VL(len("Occupation")))
  de.SetVR(gdcm.VR(gdcm.VR.SH))

  # Create an item
  it=gdcm.Item()
  it.SetVLToUndefined()      # Needed to not popup error message
  #it.InsertDataElement(de)
  nds=it.GetNestedDataSet()
  nds.Insert(de)

  # Create a Sequence
  sq=gdcm.SequenceOfItems().New()
  sq.SetLengthToUndefined()
  sq.AddItem(it)

  # Insert sequence into data set
```

```
  des=gdcm.DataElement(gdcm.Tag(0x0400,0x0550))
  des.SetVR(gdcm.VR(gdcm.VR.SQ))
  des.SetValue(sq.__ref__())
  des.SetVLToUndefined()

  ds.Insert(des)

  w = gdcm.Writer()
  w.SetFile( f )
  w.SetFileName( file2 )
  if not w.Write():
    sys.exit(1)
```

27.95 offscreenimage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkImageMapToWindowLevelColors.h"
#include "vtkImageActor.h"
#include "vtkPNGWriter.h"
#include "vtkWindowToImageFilter.h"
#include "vtkMedicalImageProperties.h"

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  const char *filename = argv[1];

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New
      ();
  reader->SetFileName( filename );
  reader->Update(); // important to read the window/level info

  vtkMedicalImageProperties *prop = reader->GetMedicalImageProperties();

  vtkRenderWindow *renWin = vtkRenderWindow::New();
  renWin->OffScreenRenderingOn();

  vtkRenderer *renderer = vtkRenderer::New();
  renWin->AddRenderer(renderer);

  vtkImageMapToWindowLevelColors *windowlevel =
      vtkImageMapToWindowLevelColors::New();
  windowlevel->SetInput( reader->GetOutput() );
  unsigned int n = prop->GetNumberOfWindowLevelPresets();
  if( n )
    {
    // Take the first one by default:
    const double *wl = prop->GetNthWindowLevelPreset(0);
    windowlevel->SetWindow( wl[0] );
    windowlevel->SetLevel( wl[1] );
    }

  vtkImageActor *actor = vtkImageActor::New();
  actor->SetInput( windowlevel->GetOutput() );

  renderer->AddActor( actor );

  renWin->Render();
```

```
  vtkWindowToImageFilter *w2if = vtkWindowToImageFilter::New();
  w2if->SetInput ( renWin );

  vtkPNGWriter *wr = vtkPNGWriter::New();
  wr->SetInput( w2if->GetOutput() );
  wr->SetFileName ( "offscreenimage.png" );
  wr->Write();

  reader->Delete();
  renWin->Delete();
  renderer->Delete();
  windowlevel->Delete();
  actor->Delete();
  w2if->Delete();
  wr->Delete();

  return 0;
}
```

27.96 PatchFile.cxx

This is a C++ example on how to use gdcm::Attribute

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * The image was a broken file where the Pixel Data element was 8 times too big
 * Apparently multiplying the BitsAllocated to 4 and multiplying the number of
 * frames by 2 would solve the problem
 *
 * This C++ code can be used to patch the header.
 */

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    return 1;
    }
  const char *f = argv[1];
  const char *out = argv[2];
  gdcm::Reader r;
  r.SetFileName( f );
  if( !r.Read() )
    {
    return 1;
    }

  gdcm::File &file = r.GetFile();
  gdcm::DataSet& ds = file.GetDataSet();
  // (0028,0100) US 16                                       #   2, 1
       BitsAllocated
  // (0028,0101) US 16                                       #   2, 1
       BitsStored
  // (0028,0102) US 15                                       #   2, 1 HighBit
  //
    {
    gdcm::Attribute<0x28,0x100> at;
    at.SetFromDataElement( ds.GetDataElement(
```

```
      at.GetTag() ) );
    if( at.GetValue() != 8 )
      {
      return 1;
      }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
    }
    {
    gdcm::Attribute<0x28,0x101> at;
    at.SetFromDataElement( ds.GetDataElement(
      at.GetTag() ) );
    if( at.GetValue() != 8 )
      {
      return 1;
      }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
    }
    {
    gdcm::Attribute<0x28,0x102> at;
    at.SetFromDataElement( ds.GetDataElement(
      at.GetTag() ) );
    if( at.GetValue() != 7 )
      {
      return 1;
      }
    at.SetValue( 31 );
    ds.Replace( at.GetAsDataElement() );
    }
  // (0028,0008) IS [56]                                    #   2, 1
       NumberOfFrames

    {
    gdcm::Attribute<0x28,0x8> at;
    at.SetFromDataElement( ds.GetDataElement(
      at.GetTag() ) );
    at.SetValue( at.GetValue() * 2 );
    ds.Replace( at.GetAsDataElement() );
    }

  gdcm::Writer w;
  w.SetFile( file );
  w.SetCheckFileMetaInformation( false );
  w.SetFileName( out );
  if( !w.Write() )
    {
    return 1;
    }

  // Now let's see if we can read it as an image:
  gdcm::ImageReader ir;
  ir.SetFileName( out );
  if(!ir.Read())
    {
    return 1;
    }
  gdcm::Image &image = ir.GetImage();
  unsigned long len = image.GetBufferLength();
  const gdcm::ByteValue *bv = ir.GetFile().GetDataSet
      ().GetDataElement( gdcm::Tag(0x7fe0,0x0010) ).
      GetByteValue();
  if( !bv || len != bv->GetLength() )
    {
    return 1;
    }
  std::cout << bv->GetLength() << " " << len << std::endl;

  std::cout << "Sucess to rewrite image !" << std::endl;
  image.Print( std::cout );
  return 0;
}
```

27.97 PhilipsPrivateRescaleInterceptSlope.py

```
##############################################################################
#
```

```
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

"""
Usage:

 python
"""

import gdcm
import sys

filename = sys.argv[1]
tmpfile = "/tmp/philips_rescaled.dcm"


# Need to access some private tags, read the file :
reader = gdcm.Reader()
reader.SetFileName( filename )
if not reader.Read():
  sys.exit(1)

ds = reader.GetFile().GetDataSet()

#print ds
# (2005,1409)     DS      4       0.0
# (2005,140a)     DS      16      1.52283272283272

# (2005,0014)     LO      26      Philips MR Imaging DD 005
tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
print tag1
print tag2

# make sure to do a copy, we want the private tag to remain
# otherwise gdcm gives us a reference
el1 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
print el1
el2 = gdcm.DataElement( ds.GetDataElement( tag2 ) )
print el2

# (0028,1052) DS [-1000]                                  #   6, 1
       RescaleIntercept
# (0028,1053) DS [1]                                      #   2, 1 RescaleSlope

el1.SetTag( gdcm.Tag(0x0028,0x1052) )
el2.SetTag( gdcm.Tag(0x0028,0x1053) )

ds.Insert( el1 )
ds.Insert( el2 )

w = gdcm.Writer()
w.SetCheckFileMetaInformation( False )
w.SetFileName( tmpfile )
w.SetFile( reader.GetFile() )
if not w.Write():
  sys.exit(1)

print "success"
```

27.98 PlaySound.py

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
```

```
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################

"""
Usage:

 python PlaySound.py input.dcm
"""

import gdcm
import sys

#filename =
       "/home/mmalaterre/Creatis/gdcmDataExtra/gdcmNonImageData/audio_from_rafael_sanguinetti.dcm"
filename = sys.argv[1]
print filename

r = gdcm.Reader()
r.SetFileName( filename )
if not r.Read():
  sys.exit(1)

ds = r.GetFile().GetDataSet()

waveformtag = gdcm.Tag(0x5400,0x0100)
waveformsq = ds.GetDataElement( waveformtag )
#print waveformsq

#print dir(waveformsq)

items = waveformsq.GetSequenceOfItems()

if not items.GetNumberOfItems():
  sys.exit(1)

item = items.GetItem(1)
#print item

waveformds = item.GetNestedDataSet()
#print waveformds

waveformdatatag = gdcm.Tag(0x5400,0x1010)
waveformdata = waveformds.GetDataElement( waveformdatatag )

#print waveformdata.GetPointer()
bv = waveformdata.GetByteValue()
print dir(bv)

#print bv.GetPointer()
print bv.GetLength()
l = 116838

file='test.wav'
myfile = open(file, "wb")
s = bv.GetPointer()
for i in range(0, l):
  myfile.write(s[i])
myfile.close()

# http://mail.python.org/pipermail/python-list/2004-October/288905.html
if sys.platform.startswith('win'):
   from winsound import PlaySound, SND_FILENAME, SND_ASYNC
   PlaySound(file, SND_FILENAME|SND_ASYNC)
elif sys.platform.find('linux')>-1:
   from wave import open as waveOpen
   from ossaudiodev import open as ossOpen
   s = waveOpen(file,'rb')
   (nc,sw,fr,nf,comptype, compname) = s.getparams( )
   dsp = ossOpen('/dev/dsp','w')
   try:
     from ossaudiodev import AFMT_S16_NE
   except ImportError:
     if byteorder == "little":
       AFMT_S16_NE = ossaudiodev.AFMT_S16_LE
     else:
       AFMT_S16_NE = ossaudiodev.AFMT_S16_BE
```

```
   dsp.setparameters(AFMT_S16_NE, nc, fr)
   data = s.readframes(nf)
   s.close()
   dsp.write(data)
   dsp.close()
```

27.99 pmsct_rgb1.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to rewrite a ELSCINT1/PMSCT_RGB1 compressed
 * image so that it is readable by most 3rd party software (DICOM does
 * not specify this particular encoding).
 * This is required for the sake of interoperability with any standard
 * conforming DICOM system.
 *
 * Everything done in this code is for the sole purpose of writing
       interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Jean-Pierre Roux for providing the sample datasets
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

void delta_decode(const unsigned char *data_in, size_t data_size,
  std::vector<unsigned char> &new_stream, unsigned short pc, size_t w, size_t h
      )
{
  const size_t plane_size = h * w;
  const size_t outputlen = 3 * plane_size;
  new_stream.resize( outputlen );

  assert( data_size != outputlen );
  if( data_size == outputlen )
    {
    return;
    }
  typedef unsigned char byte;
  enum {
    COLORMODE  = 0x81,
    ESCMODE    = 0x82,
    REPEATMODE = 0x83
  };

  byte* src = (byte*)data_in;
  byte* dest = (byte*)&new_stream[0];
  union { byte gray; byte rgb[3]; } pixel;
  pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
  // always start in grayscale mode
  bool graymode = true;
  size_t dx = 1;
  size_t dy = 3;
  // algorithm works with both planar configuration
  // It does produce surprising greenish background color for planar
  // configuration is 0, while the nested Icon SQ display a nice black
```

```
// background
if (pc)
  {
  dx = plane_size;
  dy = 1;
  }
size_t ps = plane_size;

// The following is highly unoptimized as we have nested if statement in a
     while loop
// we need to switch from one algorithm to ther other (RGB <-> GRAY)
while (ps)
  {
  // next byte:
  byte b = *src++;
  assert( src < data_in + data_size );
  // mode selection:
  switch ( b )
    {
  case ESCMODE:
    // Used to treat a byte 81/82/83 as a normal byte
    if (graymode)
      {
      pixel.gray += *src++;
      dest[0*dx] = pixel.gray;
      dest[1*dx] = pixel.gray;
      dest[2*dx] = pixel.gray;
      }
    else
      {
      pixel.rgb[0] += *src++;
      pixel.rgb[1] += *src++;
      pixel.rgb[2] += *src++;
      dest[0*dx] = pixel.rgb[0];
      dest[1*dx] = pixel.rgb[1];
      dest[2*dx] = pixel.rgb[2];
      }
    dest += dy;
    ps--;
    break;
  case REPEATMODE:
    // repeat mode (RLE)
    b = *src++;
    ps -= b;
    if (graymode)
      {
      while (b-- > 0)
        {
        dest[0*dx] = pixel.gray;
        dest[1*dx] = pixel.gray;
        dest[2*dx] = pixel.gray;
        dest += dy;
        }
      }
    else
      {
      while (b-- > 0)
        {
        dest[0*dx] = pixel.rgb[0];
        dest[1*dx] = pixel.rgb[1];
        dest[2*dx] = pixel.rgb[2];
        dest += dy;
        }
      }
    break;
  case COLORMODE:
    // We are swithing from one mode to the other. The stream contains an
     intermixed
    // compression of RGB codec and GRAY codec. Each one not knowing of the
     other
    // reset old value to 0.
    if (graymode)
      {
      graymode = false;
      pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
      }
    else
      {
      graymode = true;
      pixel.gray = 0;
      }
```

```
      break;
    default:
      // This is identical to ESCMODE, it would be nicer to use fall-through
      if (graymode)
        {
        pixel.gray += b;
        dest[0*dx] = pixel.gray;
        dest[1*dx] = pixel.gray;
        dest[2*dx] = pixel.gray;
        }
      else
        {
        pixel.rgb[0] += b;
        pixel.rgb[1] += *src++;
        pixel.rgb[2] += *src++;
        dest[0*dx] = pixel.rgb[0];
        dest[1*dx] = pixel.rgb[1];
        dest[2*dx] = pixel.rgb[2];
        }
      dest += dy;
      ps--;
      break;
      } // end switch
    } // end while
}

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet
      ();

  // (07a1,1011) CS [PMSCT_RGB1]                                    # 10,1
       Tamar Compression Type
  const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"
      ELSCINT1");
  if( !ds.FindDataElement( tcompressiontype ) ) return 1;
  const gdcm::DataElement& compressiontype = ds.GetDataElement
      ( tcompressiontype );
  if ( compressiontype.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = compressiontype.GetByteValue
      ();
  std::string comprle = "PMSCT_RLE1";
  std::string comprgb = "PMSCT_RGB1";
  bool isrle = false;
  bool isrgb = false;
  if( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) ==
       0 )
    {
    isrle = true;
    return 1;
    }
  if( strncmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) ==
       0 )
    {
    isrgb = true;
    }
  if( !isrgb && !isrle ) return 1;

  const gdcm::PrivateTag tcompressedpixeldata(0x07a1,0x000a,"
      ELSCINT1");
  if( !ds.FindDataElement( tcompressedpixeldata) ) return 1;
  const gdcm::DataElement& compressionpixeldata = ds.
      GetDataElement( tcompressedpixeldata);
  if ( compressionpixeldata.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv2 = compressionpixeldata.
      GetByteValue();

  gdcm::Attribute<0x0028,0x0006> at0;
  at0.SetFromDataSet( ds );
  gdcm::Attribute<0x0028,0x0010> at1;
  at1.SetFromDataSet( ds );
  gdcm::Attribute<0x0028,0x0011> at2;
```

```
  at2.SetFromDataSet( ds );

  std::vector<unsigned char> buffer;
  delta_decode((const unsigned char*)bv2->GetPointer(), bv2->
      GetLength(), buffer,
    at0.GetValue(), at1.GetValue(), at2.GetValue() );

  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0
      x0010) );
  pixeldata.SetVR( gdcm::VR::OW );
  pixeldata.SetByteValue( (char*)&buffer[0], buffer.size() );
  // TODO we should check that decompress byte buffer match the expected size
       (row*col*...)

  // Add the pixel data element
  reader.GetFile().GetDataSet().Replace( pixeldata );

  reader.GetFile().GetHeader().SetDataSetTransferSyntax
      (
    gdcm::TransferSyntax::ExplicitVRLittleEndian
      );
  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );

  // Cleanup stuff:
  // remove the compressed pixel data:
  // FIXME: should I remove more private tags ? all of them ?
  // oh well this is just an example
  // use gdcm::Anonymizer::RemovePrivateTags if needed...
  writer.GetFile().GetDataSet().Remove(
      compressionpixeldata.GetTag() );
  std::string outfilename;
  if (argc > 2)
     outfilename = argv[2];
  else
     outfilename = "outrgb.dcm";
  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr << "Failed to write" << std::endl;
    return 1;
    }

  std::cout << "success !" << std::endl;

  return 0;
}
```

27.100 PrivateDict.py

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

"""
"""

import gdcm
import sys,os

if __name__ == "__main__":
  #gdcm.Trace.DebugOn()
  globInst = gdcm.Global.GetInstance()
  # Try to load Part3.xml file
  # This fils is too big for being accessible directly at runtime.
  globInst.LoadResourcesFiles()
```

```
  # Get a private tag from the runtime dicts. LoadResourcesFiles could
  # have failed but this has no impact on the private dict

  d = globInst.GetDicts()
  print d.GetDictEntry( gdcm.Tag(0x0029,0x0010) ,"SIEMENS CSA HEADER" )
  pd = d.GetPrivateDict()
  print pd.GetDictEntry( gdcm.PrivateTag(0x0029,0x0010,"SIEMENS
       CSA HEADER") )
```

27.101 PublicDict.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Dummy example to show GDCM Dict(s) API (Part 6) + Collected Private
       Attributes:
 */

#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmCSAHeader.h"
#include "gdcmPrivateTag.h"

int main(int , char *[])
{
  const gdcm::Global& g = gdcm::Global::GetInstance
      (); // sum of all knowledge !
  const gdcm::Dicts &dicts = g.GetDicts();
  const gdcm::Dict &pub = dicts.GetPublicDict(); // Part
       6

  //std::cout << pub << std::endl;

  // 3 differents way to access the same information

  // 1. From the public dict only:
  gdcm::Tag patient_name(0x10,0x10);
  const gdcm::DictEntry &entry1 = pub.GetDictEntry(
      patient_name);
  std::cout << entry1 << std::endl;

  // 2. From all dicts:
  const gdcm::DictEntry &entry2 = dicts.GetDictEntry
      (patient_name);
  std::cout << entry2 << std::endl;

  // 3. This solution is the most flexible solution as you can request using
       the same
  // API either a public tag or a private tag
  const char *strowner = 0;
  const gdcm::DictEntry &entry3 = dicts.GetDictEntry
      (patient_name,strowner);
  std::cout << entry3 << std::endl;

  // Private attributes:

  // try with a private tag now:
  const gdcm::PrivateTag &private_tag =
      gdcm::CSAHeader::GetCSAImageHeaderInfoTag
      ();
  //std::cout << private_tag << std::endl;
  const gdcm::DictEntry &entry4 = dicts.GetDictEntry
      (private_tag,private_tag.GetOwner());
  std::cout << entry4 << std::endl;
```

```
  // Let's pretend that private lookup is on 0x10xx elements:
  gdcm::PrivateTag dummy = private_tag;
  dummy.SetElement( 0x1000 + dummy.GetElement() );
  const gdcm::DictEntry &entry5 = dicts.GetDictEntry
      (dummy,dummy.GetOwner());
  std::cout << entry5 << std::endl;


  return 0;
}
```

27.102 ReadAndDumpDICOMDIR.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to read and dump a DICOMDIR File
 *
 * Thanks:
 *   Tom Marynowski (lordglub gmail) for contributing this example
 */
#include "gdcmReader.h"
#include "gdcmMediaStorage.h"

typedef std::set<gdcm::DataElement> DataElementSet;
typedef DataElementSet::const_iterator ConstIterator;

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];

  gdcm::Reader reader;
  reader.SetFileName( filename);
  if( !reader.Read() )
    {
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
    }
  std::stringstream strm;

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  gdcm::FileMetaInformation &fmi = file.GetHeader
      ();

  gdcm::MediaStorage ms;
  ms.SetFromFile(file);
  if( ms != gdcm::MediaStorage::MediaStorageDirectoryStorage
       )
    {
    std::cout << "This file is not a DICOMDIR" << std::endl;
    return 1;
    }

  if (fmi.FindDataElement( gdcm::Tag (0x0002, 0x0002)))
    {   strm.str("");
    fmi.GetDataElement( gdcm::Tag (0x0002, 0x0002) ).
      GetValue().Print(strm);
    }
  else
    {
    std::cerr << " Media Storage Sop Class UID not present" << std::cout;
    }
```

```
//TODO il faut trimer strm.str() avant la comparaison au cas ou...
if ("1.2.840.10008.1.3.10"!=strm.str())
  {
  std::cout << "This file is not a DICOMDIR" << std::endl;
  return 1;
  }

ConstIterator it = ds.GetDES().begin();

for( ; it != ds.GetDES().end(); ++it)
  {

  if (it->GetTag()==gdcm::Tag (0x0004, 0x1220))
    {

    const gdcm::DataElement &de = (*it);
    // ne pas utiliser GetSequenceOfItems pour extraire les items
    gdcm::SmartPointer<gdcm::SequenceOfItems>
     sqi =de.GetValueAsSQ();
    unsigned int itemused = 1;
    while (itemused<=sqi->GetNumberOfItems())

      {
      strm.str("");

      if (sqi->GetItem(itemused).FindDataElement(
    gdcm::Tag (0x0004, 0x1430)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag
     (0x0004, 0x1430)).GetValue().Print(strm);

      //TODO il faut trimer strm.str() avant la comparaison
      while((strm.str()=="PATIENT")||((strm.str()=="PATIENT ")))
        {
        std::cout << strm.str() << std::endl;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
    gdcm::Tag (0x0010, 0x0010)))
          sqi->GetItem(itemused).GetDataElement(
    gdcm::Tag (0x0010, 0x0010)).GetValue().Print(strm);
        std::cout << "PATIENT NAME : " << strm.str() << std::endl;


        //PATIENT ID
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
    gdcm::Tag (0x0010, 0x0020)))
          sqi->GetItem(itemused).GetDataElement(
    gdcm::Tag (0x0010, 0x0020)).GetValue().Print(strm);
        std::cout << "PATIENT ID : " << strm.str() << std::endl;

        /*ADD TAG TO READ HERE*/
        std::cout << "========================== "  << std::endl;
        itemused++;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
    gdcm::Tag (0x0004, 0x1430)))
          sqi->GetItem(itemused).GetDataElement(
    gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

        //TODO il faut trimer strm.str() avant la comparaison
        while((strm.str()=="STUDY")||((strm.str()=="STUDY ")))
          {
          std::cout << "  " << strm.str() << std::endl;
          //UID
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(
    gdcm::Tag (0x0020, 0x000d)))
            sqi->GetItem(itemused).GetDataElement(
    gdcm::Tag (0x0020, 0x000d)).GetValue().Print(strm);
          std::cout << "      STUDY UID : " << strm.str() << std::endl;

          //STUDY DATE
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(
    gdcm::Tag (0x0008, 0x0020)))
            sqi->GetItem(itemused).GetDataElement(
    gdcm::Tag (0x0008, 0x0020)).GetValue().Print(strm);
          std::cout << "      STUDY DATE : " << strm.str() << std::endl;

          //STUDY DESCRIPTION
          strm.str("");
```

```
      if (sqi->GetItem(itemused).FindDataElement(
gdcm::Tag (0x0008, 0x1030)))
        sqi->GetItem(itemused).GetDataElement(
gdcm::Tag (0x0008, 0x1030)).GetValue().Print(strm);
      std::cout << "      STUDY DESCRIPTION : " << strm.str() <<
std::endl;

      /*ADD TAG TO READ HERE*/
      std::cout << "      " << "========================== "  <<
std::endl;

      itemused++;
      strm.str("");
      if (sqi->GetItem(itemused).FindDataElement(
gdcm::Tag (0x0004, 0x1430)))
        sqi->GetItem(itemused).GetDataElement(
gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

      //TODO il faut trimer strm.str() avant la comparaison
      while((strm.str()=="SERIES")|||((strm.str()=="SERIES ")))
        {
        std::cout << "      " << strm.str() << std::endl;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcm::Tag (0x0020, 0x000e)))
          sqi->GetItem(itemused).GetDataElement(
gdcm::Tag (0x0020, 0x000e)).GetValue().Print(strm);
        std::cout << "          SERIE UID" << strm.str() << std::endl;

        //SERIE MODALITY
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcm::Tag (0x0008, 0x0060)))
          sqi->GetItem(itemused).GetDataElement(
gdcm::Tag (0x0008, 0x0060)).GetValue().Print(strm);
        std::cout << "          SERIE MODALITY" << strm.str() <<
std::endl;

        //SERIE DESCRIPTION
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcm::Tag (0x0008, 0x103e)))
          sqi->GetItem(itemused).GetDataElement(
gdcm::Tag (0x0008, 0x103e)).GetValue().Print(strm);
        std::cout << "          SERIE DESCRIPTION" << strm.str() <<
std::endl;


        /*ADD TAG TO READ HERE*/

        std::cout << "          " << "========================== "  <<
std::endl;
        itemused++;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcm::Tag (0x0004, 0x1430)))
          sqi->GetItem(itemused).GetDataElement(
gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);


        //TODO il faut trimer strm.str() avant la comparaison
        while ((strm.str()=="IMAGE")|||((strm.str()=="IMAGE ")))
          // if(tmp=="IMAGE")
          {
          std::cout << "          " << strm.str() << std::endl;


          //UID
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement
(gdcm::Tag (0x0004, 0x1511)))
            sqi->GetItem(itemused).GetDataElement(
gdcm::Tag (0x0004, 0x1511)).GetValue().Print(strm);
          std::cout << "              IMAGE UID : " << strm.str() <<
std::endl;

          //PATH de l'image
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement
(gdcm::Tag (0x0004, 0x1500)))
            sqi->GetItem(itemused).GetDataElement(
```

```
      gdcm::Tag (0x0004, 0x1500)).GetValue().Print(strm);
                std::cout << "              IMAGE PATH : " << strm.str() <<
      std::endl;
                /*ADD TAG TO READ HERE*/



                if(itemused < sqi->GetNumberOfItems())
                  {itemused++;
                  }else{break;}

                strm.str("");

                if (sqi->GetItem(itemused).FindDataElement
      (gdcm::Tag (0x0004, 0x1430)))
                  sqi->GetItem(itemused).GetDataElement(
      gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

                }
              }
            }
          }
        itemused++;
        }
      }
    }
  return 0;
}
```

27.103 ReadAndDumpDICOMDIR.py

```
      ##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#  File: ReadAndDumpDICOMDIR.py
#
#  Author: Lukas Batteau (lbatteau gmail)
#
#  This example shows how to read and dump a DICOMDIR File.
#  Based on Tom Marynowski's (lordglub gmail) example.
#
#  Usage:
#  python ReadAndDumpDICOMDIR.py [DICOMDIR file]
##############################################################################


import sys
import gdcm

if __name__ == "__main__":
    # Check arguments
    if (len(sys.argv) < 2):
        # No filename passed
        print "No input filename found"
        quit()

    filename = sys.argv[1]


    # Read file
    reader = gdcm.Reader()
    reader.SetFileName(filename)
    if (not reader.Read()):
        print "Unable to read %s" % (filename)
        quit()
```

```
file = reader.GetFile()

# Retrieve header information
fileMetaInformation = file.GetHeader()
print fileMetaInformation

# Retrieve data set
dataSet = file.GetDataSet()
#print dataSet

# Check media storage
mediaStorage = gdcm.MediaStorage()
mediaStorage.SetFromFile(file)
if (gdcm.MediaStorage.GetMSType(str(mediaStorage
  )) != gdcm.MediaStorage.MediaStorageDirectoryStorage):
    # File is not a DICOMDIR
    print "This file is not a DICOMDIR (Media storage type: %s)" % (str(
  mediaStorage))
    quit()

# Check Media Storage SOP Class
if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))):
    sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0
  x0002, 0x0002)).GetValue())
    # Check SOP UID
    if (sopClassUid != "1.2.840.10008.1.3.10"):
        # File is not a DICOMDIR
        print "This file is not a DICOMDIR"
else:
    # Not present
    print "Media Storage SOP Class not present"
    quit()

# Iterate through the DICOMDIR data set
iterator = dataSet.GetDES().begin()
while (not iterator.equal(dataSet.GetDES().end())):
    dataElement = iterator.next()

    # Check the element tag
    if (dataElement.GetTag() == gdcm.Tag(0x004, 0x1220)):
        # The 'Directory Record Sequence' element
        sequence = dataElement.GetValueAsSQ()

        # Loop through the sequence items
        itemNr = 1
        while (itemNr < sequence.GetNumberOfItems()):
            item = sequence.GetItem(itemNr)

            # Check the element tag
            if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
                # The 'Directory Record Type' element
                value = str(item.GetDataElement(gdcm.Tag(0x0004, 0
  x1430)).GetValue())

                # PATIENT
                while (value.strip() == "PATIENT"):
                    print value.strip()
                    # Print patient name
                    if (item.FindDataElement(gdcm.Tag(0x0010, 0
  x0010))):
                        value = str(item.GetDataElement(gdcm.Tag(0
  x0010, 0x0010)).GetValue())
                        print value

                    # Print patient ID
                    if (item.FindDataElement(gdcm.Tag(0x0010, 0
  x0020))):
                        value = str(item.GetDataElement(gdcm.Tag(0
  x0010, 0x0020)).GetValue())
                        print value

                    # Next
                    itemNr = itemNr + 1
                    item = sequence.GetItem(itemNr)
                    if (item.FindDataElement(gdcm.Tag(0x0004, 0
  x1430))):
                        value = str(item.GetDataElement(gdcm.Tag(0
  x0004, 0x1430)).GetValue())

                    # STUDY
                    while (value.strip() == "STUDY"):
```

```
                        print value.strip()

                        # Print study UID
                        if (item.FindDataElement(gdcm.Tag(0x0020, 0
x000d))):
                            value = str(item.GetDataElement(gdcm.Tag
(0x0020, 0x000d)).GetValue())
                            print value

                        # Print study date
                        if (item.FindDataElement(gdcm.Tag(0x0008, 0
x0020))):
                            value = str(item.GetDataElement(gdcm.Tag
(0x0008, 0x0020)).GetValue())
                            print value

                        # Print study description
                        if (item.FindDataElement(gdcm.Tag(0x0008, 0
x1030))):
                            value = str(item.GetDataElement(gdcm.Tag
(0x0008, 0x1030)).GetValue())
                            print value

                        # Next
                        itemNr = itemNr + 1
                        item = sequence.GetItem(itemNr)
                        if (item.FindDataElement(gdcm.Tag(0x0004, 0
x1430))):
                                value = str(item.GetDataElement(gdcm.Tag
(0x0004, 0x1430)).GetValue())

                        # SERIES
                        while (value.strip() == "SERIES"):
                            print value.strip()

                            # Print series UID
                            if (item.FindDataElement(gdcm.Tag(0
x0020, 0x000e))):
                                value = str(item.GetDataElement(gdcm.Tag
(0x0020, 0x000e)).GetValue())
                                print value

                            # Print series modality
                            if (item.FindDataElement(gdcm.Tag(0
x0008, 0x0060))):
                                value = str(item.GetDataElement(gdcm.Tag
(0x0008, 0x0060)).GetValue())
                                print "Modality"
                                print value

                            # Print series description
                            if (item.FindDataElement(gdcm.Tag(0
x0008, 0x103e))):
                                value = str(item.GetDataElement(gdcm.Tag
(0x0008, 0x103e)).GetValue())
                                print "Description"
                                print value

                            # Next
                            itemNr = itemNr + 1
                            item = sequence.GetItem(itemNr)
                            if (item.FindDataElement(gdcm.Tag(0
x0004, 0x1430))):
                                value = str(item.GetDataElement(gdcm.Tag
(0x0004, 0x1430)).GetValue())

                            # IMAGE
                            while (value.strip() == "IMAGE"):
                                print value.strip()

                                # Print image UID
                                if (item.FindDataElement(gdcm.Tag(0
x0004, 0x1511))):
                                    value = str(item.GetDataElement(
gdcm.Tag(0x0004, 0x1511)).GetValue())
                                    print value

                                # Next
                                if (itemNr < sequence.GetNumberOfItems()):
                                    itemNr = itemNr + 1
                                else:
```

```
                                        break

                                    item = sequence.GetItem(itemNr)
                                    if (item.FindDataElement(gdcm.Tag(0
      x0004, 0x1430))):
                                            value = str(item.GetDataElement(
      gdcm.Tag(0x0004, 0x1430)).GetValue())

                # Next
                itemNr = itemNr + 1
```

27.104 ReadAndPrintAttributes.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This small example will show how one can read and print
 * a DICOM Attribute using different technique (by tag or by name)
 */

#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmAttribute.h"
#include "gdcmStringFilter.h"

#include <iostream>

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr << argv[0] << " input.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];

  // Instanciate the reader:
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
    }

  // The output of gdcm::Reader is a gdcm::File
  gdcm::File &file = reader.GetFile();

  // the dataset is the the set of element we are interested in:
  gdcm::DataSet &ds = file.GetDataSet();

  const gdcm::Global& g = gdcm::Global::GetInstance
      ();
  const gdcm::Dicts &dicts = g.GetDicts();
  const gdcm::Dict &pubdict = dicts.GetPublicDict();

  using namespace gdcm;

  // In this example we will show why using name to lookup attribute can be
  // dangerous.
  Tag tPatientName(0x0,0x0);
  //const DictEntry &de1 =
  pubdict.GetDictEntryByName("Patient Name", tPatientName);
```

```
  std::cout << "Found: " << tPatientName << std::endl;

  // Indeed the attribute could not be found. Since DICOM 2003, Patient Name
  // has become Patient's Name.

  Tag tPatientsName;
  //const DictEntry &de2 =
  pubdict.GetDictEntryByName("Patient's Name", tPatientsName)
      ;

  std::cout << "Found: " << tPatientsName << std::endl;

  // Let's try to read an arbitrary DICOM Attribute:
  Tag tDoseGridScaling;
  //const DictEntry &de3 =
  pubdict.GetDictEntryByName("Dose Grid Scaling",
      tDoseGridScaling);

  std::cout << "Found: " << tDoseGridScaling << std::endl;

  if( ds.FindDataElement( tDoseGridScaling ) )
    {
    gdcm::StringFilter sf;
    sf.SetFile(file);
    std::cout << "Attribute Value as String: " << sf.ToString(
      tDoseGridScaling ) << std::endl;

    // Let's check the name again:
    std::pair<std::string, std::string> pss
      = sf.ToStringPair( tDoseGridScaling );
    std::cout << "Attribute Name Checked: " << pss.first << std::endl;
    std::cout << "Attribute Value (string): " << pss.second << std::endl;

    //const DataElement &dgs = ds.GetDataElement( tDoseGridScaling );

    // Let's assume for a moment we knew the tag number:
    Attribute<0x3004,0x000e> at;
    assert( at.GetTag() == tDoseGridScaling );
    at.SetFromDataSet( ds );
    // For the sake of long term maintenance, we will not write
    // that this particular attribute is stored as a double. What if
    // a user made a mistake. It is much safer to rely on GDCM internal
    // mechanism to deduce the VR::DS type (represented as a ieee double)
    Attribute<0x3004,0x000e>::ArrayType v =
      at.GetValue();
    std::cout << "DoseGridScaling=" << v << std::endl;
    }

  return 0;
}
```

27.105 ReadExplicitLengthSQIVR.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImplicitDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmByteValue.h"
#include "gdcmSequenceOfItems.h"

using namespace gdcm;

int main(int argc, char *argv[])
```

```
{
  const char *filename = argv[1];
  gdcm::Reader r;
  r.SetFileName( filename );
  r.Read();


  //gdcm::PrivateTag pt(0xe1,0x42,"ELSCINT1");
  //gdcm::Tag pt(0x88,0x200);
  gdcm::Tag pt(0x8,0x1140);
  DataSet &ds = r.GetFile().GetDataSet();
  const DataElement &de = ds.GetDataElement( pt );

  std::cout << de << std::endl;
  const ByteValue *bv = de.GetByteValue();
  SmartPointer<SequenceOfItems> sqi = new
      SequenceOfItems;
  sqi->SetLength( bv->GetLength() );
  std::stringstream ss;
  ss.str( std::string( bv->GetPointer(), bv->GetLength() ) )
      ;
  sqi->Read<ImplicitDataElement,SwapperNoOp>(
       ss );

  std::cout << *sqi << std::endl;

  return 0;
}
```

27.106 ReadFiles.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import gdcm.*;
import java.io.File;

public class ReadFiles
{
  static int i = 0;
  public static void process(String path)
    {
    //String path = file.getPath();
    assert PosixEmulation.FileExists(path) : "Problem converting to 8bits";

    System.out.println("Reading: " + path );
    System.out.println("File: " + i++);
    Reader r = new Reader();
    try
      {
      r.SetFileName( path );
      TagSetType skip = new TagSetType();
      skip.insert( new Tag(0x7fe0,0x10) );
      boolean b = r.ReadUpToTag( new Tag(0x88,0x200), skip );
      //System.out.println("DS:\n" + r.GetFile().GetDataSet().toString() );
      }
    finally
      {
      r.delete(); // will properly call C++ destructor and close file
       descriptor
      }
    }

  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
```

```
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }

  public static void waiting (int n)
    {
    long t0, t1;
    t0 =  System.currentTimeMillis();
    do
      {
      t1 = System.currentTimeMillis();
      }
    while ((t1 - t0) < (n * 1000));
    }

  public static void main(String[] args) throws Exception
    {
    String directory = args[0];

    Directory gdir = new Directory();
    long n = gdir.Load( directory, true );
    System.out.println( gdir.toString() );
    FilenamesType files = gdir.GetFilenames();
    for( long i = 0; i < n; ++i )
      {
      String path = files.get( (int)i );
      process( path );
      }

    System.out.println( "Java API" );

    //waiting( 10 );
    for( int i = 0; i < 2; ++i )
      {
      File dir = new File(directory);
      visitAllFiles(dir);
      }
    }
}
```

27.107 ReadGEMSSDO.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"

#include <iostream>
#include <string>

using namespace gdcm;

struct SDOElement
{
  typedef std::vector<std::string>::size_type SizeType;
  const char *GetData(SizeType index) const {
```

```
    return Data[index].c_str();
  }
  SizeType GetNumberOfData() const {
    return Data.size();
  }
  void SetData(SizeType index, const char *data) {
    Data[index] = data;
  }
  const char *GetDataFormat() const {
    return DataFormat.c_str();
  }
  void SetDataFormat(const char *dataformat, SizeType num) {
    DataFormat = dataformat;
    Data.resize( num );
  }
  void Print( std::ostream &os ) const {
    os << DataFormat << ":" << std::endl;
    std::vector<std::string>::const_iterator it = Data.begin();
    size_t s = 0;
    for( ; it != Data.end(); ++it )
      {
      os << "  (" << s++ << ") " << *it << std::endl;
      }
  }
private:
  std::string DataFormat;
  std::vector<std::string> Data;
};

class SDOHeader
{
public:
  typedef std::vector<SDOElement> SDOElements;
  typedef SDOElements::size_type SizeType;
  SizeType GetNumberOfSDOElements() const {
  return InternalSDODataSet.size();
  }
  void AddSDOElement(SDOElement const &sdoelement) {
  InternalSDODataSet.push_back( sdoelement );
  }
  const SDOElement &GetSDOElement(SizeType index) const {
  return InternalSDODataSet[index];
  }
  const SDOElement &GetSDOElementByName(const char *name) const {
  return InternalSDODataSet[0];
  }
  void LoadFromAttributes(std::string const &s1, std::string const &s2)
    {
  std::string tok;
  std::string tok2;
  std::stringstream strstr(s1);
  std::stringstream strstr2(s2);

  SDOElement element;
  // Do format
  size_t count = 0;
  while ( std::getline ( strstr2, tok, '\\' ) )
    {
    //std::cout << tok << " ";
    std::getline ( strstr2, tok2, '\\' );
    //std::cout << tok2 << std::endl;
    count += atoi( tok2.c_str() );
    element.SetDataFormat( tok.c_str(), atoi( tok2.c_str() ) );
    for( size_t t = 0; t < element.GetNumberOfData(); ++t )
      {
      std::getline ( strstr, tok, '\\' );
      element.SetData(t, tok.c_str() );
      }
    AddSDOElement( element );
    }
  //while ( std::getline ( strstr, tok, '^' ) )
//  while ( std::getline ( strstr, tok, '\\' ) )
//    {
//    std::cout << tok << std::endl;
//    count++;
//    }
//  std::cout << "Count: " << count << std::endl;
//  count = 0;

//  std::cout << "Count: " << count << std::endl;
```

```
    }
  void Print( std::ostream &os ) const {
    SDOElements::const_iterator it = InternalSDODataSet.begin();
    for( ; it != InternalSDODataSet.end(); ++it )
      {
      it->Print ( os );
      }
  }
private:
  SDOElements InternalSDODataSet;
};

bool sdo_decode( DataElement const &stringdata, DataElement
       const &stringdataformat )
{
  const char *sd = stringdata.GetByteValue()->GetPointer(
      );
  const size_t len_sd = stringdata.GetByteValue()->GetLength
      ();

  std::string s1 = std::string( sd, len_sd );

  const char *sdf = stringdataformat.GetByteValue()->GetPointer
      ();
  const size_t len_sdf = stringdataformat.GetByteValue()->GetLength
      ();

  std::string s2 = std::string( sdf, len_sdf );

//  std::cout << s1 << std::endl;
//  std::cout << s2 << std::endl;

  SDOHeader header;
  header.LoadFromAttributes( s1, s2 );

  header.Print( std::cout );

  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr << argv[0] << " input.dcm" << std::endl;
    return 1;
    }
  const char *filename = argv[1];
  Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  File &file = reader.GetFile();
  DataSet &ds = file.GetDataSet();

  // StringData (0033,xx1F) 3 "GEMS_GENIE_1" List of SDO parameters stored as
  // list of strings
  const PrivateTag tstringdata(0x33,0x1f,"GEMS_GENIE_1");
  // StringDataFormat (0033,xx23) 3 "GEMS_GENIE_1" Format of string parameters;
  // contains information about name and number of strings in list
  const PrivateTag tstringdataformat(0x33,0x23,"GEMS_GENIE_1");

  if( !ds.FindDataElement( tstringdata ) ) return 1;
  const DataElement& stringdata = ds.GetDataElement(
      tstringdata );
  if( !ds.FindDataElement( tstringdataformat ) ) return 1;
  const DataElement& stringdataformat = ds.GetDataElement
      ( tstringdataformat );

  sdo_decode( stringdata, stringdataformat );

  return 0;
}
```

27.108 ReadMultiTimesException.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// The intention of this sample program is to provoke bad_alloc exceptions in
       gdcm code

#include "gdcmImageReader.h"

int main(int argc, char* argv[])
{
  // We pre-allocate some memory (about 1Gb) to help the issue to show up
       earlier
  char *dummyBuffer = new char[1024*1024*1100]; (void)dummyBuffer;
  // Check the number of parameters given
  if (argc < 3)
    {
    std::cerr << "Usage: " << argv[0] << " Filename numberOfTries" << std::endl
      ;
    return 1;
    }

  std::cout << "We are going to read the file: " << argv[1] << " " << argv[2] <
      < " times" << std::endl;
  // We hold the pointers in an array to avoid the memory to be released
  // We read the input file n-times
  for (int i = 0; i < atoi(argv[2]); ++i)
    {
    gdcm::ImageReader reader;
    std::cout << "Reading try: " << i << std::endl;
    // Read files
    reader.SetFileName(argv[1]);
    try
      {
      reader.Read();
      gdcm::Image & img = reader.GetImage();
      unsigned long len = img.GetBufferLength();
      char *buffer = new char[ len ];
      img.GetBuffer( buffer ); // do NOT de-allocate buffer !
      }
    catch (std::bad_alloc)
      {
      std::cerr << "BAD ALLOC Exception caught!" << std::endl;
      }
    catch (...)
      {
      std::cerr << "Exception caught!" << std::endl;
      }
    }

  return 0;
}
```

27.109 ReadSeriesIntoVTK.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// We are required to call the package 'vtk' eventhough I (MM) would have
       prefered
// an import statement along the line of:
// import vtkgdcm.*;
import vtk.*;

/*
 * Usage:
 * export LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:.
 * java -classpath `pwd`/vtkgdcm.jar:/usr/share/java/vtk.jar:.
       ReadSeriesIntoVTK
 */
public class ReadSeriesIntoVTK
{
  static {
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkgdcmJava");
    try {
      System.loadLibrary("vtkRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkHybridJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkVolumeRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkVolumeRendering, skipping...");
    }
  }

  public static void main(String[] args)
    {
    vtkFileOutputWindow outWin = new vtkFileOutputWindow();
    outWin.SetInstance(outWin);
    outWin.SetFileName("MVSVTKViewer.log");

    // See: http://review.source.kitware.com/#change,888
    // vtkWrapJava does not handle static keyword
    // String directory = vtkGDCMTesting.GetGDCMDataRoot();
    vtkGDCMTesting t = new vtkGDCMTesting();
    String directory = t.GetGDCMDataRoot();
    String file0 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm";
    String file1 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm";
    String file2 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm";
    String file3 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm";

    vtkStringArray s = new vtkStringArray();
    System.out.println("adding : " + file0 );
    s.InsertNextValue( file0 );
    s.InsertNextValue( file1 );
    s.InsertNextValue( file2 );
    s.InsertNextValue( file3 );

    vtkGDCMImageReader reader = new vtkGDCMImageReader
      ();
    reader.SetFileNames( s );
    reader.Update();

    System.out.println("Success reading: " + file0 );

    vtkMetaImageWriter writer = new vtkMetaImageWriter();
    writer.DebugOn();
    writer.SetCompression( false );
    writer.SetInput( reader.GetOutput() );
    writer.SetFileName( "ReadSeriesIntoVTK.mhd" );
    writer.Write();

    System.out.println("Success writing: " + writer.GetFileName() );
    }
}
```

27.110 ReadUTF8QtDir.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * GDCM API expect a const char * as input for SetFileName
 * In order to use this API from Qt, here is a simple test that
 * shows how to do it in a portable manner:
 *
 * http://doc.qt.nokia.com/latest/qdir.html#navigation-and-directory-operations
 */

#include "gdcmReader.h"
#include "gdcmDirectory.h"

#include <QDir>
#include <QString>
#include <QCoreApplication>

#include <string>
#include <fstream>

#include <stdio.h> // fopen

static int TestBothFuncs(const char *info , const char *ba_str)
{
  int res = 0;
  FILE *f = fopen( ba_str, "r" );
  if( f )
    {
    std::cout << info << " fopen: " << ba_str << std::endl;
    fclose(f);
    ++res;
    }
  gdcm::Reader reader;
  std::ifstream is( ba_str );
  if( is.is_open() )
    {
    std::cout << info << " is_open: " << ba_str << std::endl;
    ++res;
    }
  reader.SetStream( is );
  if( reader.CanRead() == true )
    {
    std::cout << info << " SetStream/CanRead:" << ba_str << std::endl;
    ++res;
    }
  is.close();
  reader.SetFileName( ba_str );
  if( reader.CanRead() == true )
    {
    std::cout << info << " SetFileName/CanRead:" << ba_str << std::endl;
    ++res;
    }
  return 4 - res;
}

static int scanFolder(const char dirname[])
{
  int res = 0;
  gdcm::Directory dir;
  unsigned int nfiles = dir.Load( dirname, true );
  const gdcm::Directory::FilenamesType &filenames
       = dir.GetFilenames();

  for( unsigned int i = 0; i < nfiles; ++i )
    {
    const char *ba_str = filenames[i].c_str();
    res += TestBothFuncs("GDCM",ba_str);
```

```
    }
  return res;
}

static int scanFolderQt(QDir const &dir, QStringList& files)
{
  int res = 0;
  QFileInfoList children = dir.entryInfoList(QDir::AllEntries|
      QDir::NoDotAndDotDot);
  for ( int i=0; i<children.count(); i++ ) {
    QFileInfo file = children.at(i);
    if ( file.isDir() == true ) {
      res += scanFolderQt(QDir(file.absoluteFilePath()), files);
      continue;
    }
    // Convert back from the internal representation to 8bits
    // toLocal8Bit() returns by copy. Need to store explicitely the QByteArray
    QByteArray str = file.absoluteFilePath().toLocal8Bit();
    const char *ba_str1 = str.constData();
    res += TestBothFuncs("QString", ba_str1);
    }
  return res;
}

int main(int argc, char *argv[])
{
  // very important:
  QCoreApplication qCoreApp( argc , argv );
  if( argc < 2 )
    {
    std::cerr << argv[0] << " dir " << std::endl;
    return 1;
    }

  int res = 0;
  const char *dirname = argv[1];
  res += scanFolder( dirname );

  QDir dir( QString::fromLocal8Bit(dirname) );
  QStringList files;
  res += scanFolderQt( dir, files);

  if( res )
    std::cerr << "Problem with UTF-8" << std::endl;
  else
    std::cerr << "Success with UTF-8" << std::endl;

  return res;
}
```

27.111 RefCounting.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using vtkgdcm;


/*
 * this is not so much an example but simply a test to make sure cstor / dstor
       work as expected
 * and call the ::New and ->Delete() of VTK style.
 */
public class RefCounting
{
  public static int Main(string[] args)
```

```
    {
    vtkGDCMTesting testing1 = vtkGDCMTesting.New
      ();
    vtkGDCMTesting testing2 = new vtkGDCMTesting();
       // just in case people do not read STYLE documentation

    vtkGDCMImageReader reader1 = vtkGDCMImageReader
      .New();
    vtkGDCMImageReader reader2 = new vtkGDCMImageReader
      ();

    vtkGDCMImageWriter writer1 = vtkGDCMImageWriter
      .New();
    vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter
      ();

    using (vtkGDCMTesting testing3 = new vtkGDCMTesting
      ())
      {
      System.Console.Write( "GetReferenceCount: " + testing1.GetReferenceCount(
      ) + "\n");
      System.Console.Write( "GetReferenceCount: " + testing2.GetReferenceCount(
      ) + "\n");
      System.Console.Write( "GetReferenceCount: " + testing3.GetReferenceCount(
      ) + "\n");
      }

    using (vtkGDCMImageReader reader3 = new
      vtkGDCMImageReader())
      {
      System.Console.Write( "GetReferenceCount: " + reader3.GetReferenceCount()
       + "\n");
      }

    using (vtkGDCMImageWriter writer3 = vtkGDCMImageWriter
      .New())
      {
      System.Console.Write( "GetReferenceCount: " + writer3.GetReferenceCount()
       + "\n");
      }

    // C# destructor will call ->Delete on all C++ object as expected.
    return 0;
    }
}
```

27.112 ReformatFile.cs

This is a C++ example on how to use gdcm::FileDerivation

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Simple C# example
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ReformatFile.exe input.dcm output.dcm
 */
using System;
using gdcm;

public class ReformatFile
{
```

```
  public static int Main(string[] args)
    {
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Reformat App"
       );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot()
       );

    string filename = args[0];
    string outfilename = args[1];

    Reader reader = new Reader();
    reader.SetFileName( filename );
    if( !reader.Read() )
      {
      System.Console.WriteLine( "Could not read: " + filename );
      return 1;
      }

    UIDGenerator uid = new UIDGenerator(); // helper for uid generation
    FileDerivation fd = new FileDerivation();
    // For the pupose of this execise we will pretend that this image is
       referencing
    // two source image (we need to generate fake UID for that).
    string ReferencedSOPClassUID = "1.2.840.10008.5.1.4.1.1.7"; // Secondary
       Capture
    fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
    fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

    // Again for the purpose of the exercise we will pretend that the image is
       a
    // multiplanar reformat (MPR):
    // CID 7202 Source Image Purposes of Reference
    // {"DCM",121322,"Source image for image processing operation"},
    fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
    // CID 7203 Image Derivation
    // { "DCM",113072,"Multiplanar reformatting" },
    fd.SetDerivationCodeSequenceCodeValue( 113072 );
    fd.SetFile( reader.GetFile() );
    // If all Code Value are ok the filter will execute properly
    if( !fd.Derive() )
      {
      return 1;
      }

    gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

    Writer writer = new Writer();
    writer.SetFileName( outfilename );
    writer.SetFile( fd.GetFile() );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write: " + outfilename );
      return 1;
      }


    return 0;
    }
}
```

27.113 RemovePrivateTags.py

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

"""
Usage:

 python RemovePrivateTags.py input.dcm output.dcm
"""

import sys
import gdcm


if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  # Instanciate the reader.
  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  # Remove private tags
  ano = gdcm.Anonymizer()
  ano.SetFile( r.GetFile() )
  if not ano.RemovePrivateTags():
    sys.exit(1)

  # Write DICOM file
  w = gdcm.Writer()
  w.SetFile( ano.GetFile() )
  #w.CheckFileMetaInformationOff() # Do not attempt to check meta header
  w.SetFileName( file2 )
  if not w.Write():
    sys.exit(1)

  # It is usually a good idea to exit the script with an error, as gdcm does
      not remove partial (incorrect) DICOM file
  # (application level)
```

27.114 RescaleImage.cs

```
/*=============================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=============================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressImage.exe gdcmData/012345.002.050.dcm rescaled.dcm
 */
using System;
using gdcm;

public class DecompressImage
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
```

```
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    Image image = reader.GetImage();
    PixelFormat pixeltype = image.GetPixelFormat();

    Rescaler r = new Rescaler();
    r.SetIntercept( 0 );
    r.SetSlope( 1.2 );
    r.SetPixelFormat( pixeltype );
    PixelFormat outputpt = new PixelFormat( r.ComputeInterceptSlopePixelType()
      );

    System.Console.WriteLine( "pixeltype" );
    System.Console.WriteLine( pixeltype.toString() );
    System.Console.WriteLine( "outputpt" );
    System.Console.WriteLine( outputpt.toString() );

    uint len = image.GetBufferLength();
    short[] input = new short[ len / 2 ]; // sizeof(short) == 2
    image.GetArray( input );

    double[] output = new double[ len / 2 ];
    r.Rescale( output, input, len );

    // First Pixel is:
    System.Console.WriteLine( "Input:" );
    System.Console.WriteLine( input[0] );

    System.Console.WriteLine( "Output:" );
    System.Console.WriteLine( output[0] );

    return 0;
    }
}
```

27.115 reslicesphere.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
//
// Load a DICOM series.
// Position a sphere within the volume.
// Allow the user to change between Axial, Sagittal, Coronal, and
//     Oblique view of the images and move through the slices.
// The display should show the resliced image and the cross section
// of the sphere intersecting that plane.
//


/*
from  Scott Johnson /Scott Johnson neuwave com/
to  VTK /vtkusers vtk.org/
date  Tue, May 11, 2010 at 7:01 PM
*/
#include <strstream>
#include <string>

#include <vtkDICOMImageReader.h>
#include <vtkStringArray.h>
#include <vtkDirectory.h>
#include <vtkImageThreshold.h>
#include <vtkImageShiftScale.h>
```

```
#include <vtkImageReslice.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageViewer2.h>
#include <vtkSphereSource.h>
#include <vtkPolyDataMapper.h>
#include <vtkPlane.h>
#include <vtkCutter.h>
#include <vtkActor.h>
#include <vtkCommand.h>
#include <vtkSmartPointer.h>
#include <vtkMatrix4x4.h>
#include <vtkInteractorObserver.h>
#include <vtkProperty.h>
#include <vtkRenderer.h>
#include <vtkImageData.h>
#include <vtkImageActor.h>
#include "vtkTransformPolyDataFilter.h"
#include <vtkCamera.h>
#include <vtkMath.h>
#include <vtkTransform.h>
#include <vtkTextActor.h>
#include <vtkActor2D.h>
#include <vtkPolyDataMapper2D.h>
#include <vtkProperty2D.h>
#include <vtkGDCMImageReader.h>
#include <vtkImageChangeInformation.h>

#include "gdcmDirectory.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"

// Change to match the path to find Raw_0.vti or provide
// the parameter when starting ResliceSphere.

const double sphereCenter[3]={74, 219, 70};

// Angles (0, 0, 0)
const double AxialMatrix[]    = { 1.0,  0.0,  0.0,  0.0,
                                  0.0,  1.0,  0.0,  0.0,
                                  0.0,  0.0,  1.0,  0.0,
                                  0.0,  0.0,  0.0,  1.0 };
// Angles (0, 90, 0)
const double SagittalMatrix[] = { 0.0,  0.0,  1.0,  0.0,
                                  0.0,  1.0,  0.0,  0.0,
                                 -1.0,  0.0,  0.0,  0.0,
                                  0.0,  0.0,  0.0,  1.0 };
// Angles (-90, 0, 0)
const double CoronalMatrix[]  = { 1.0,  0.0,  0.0,  0.0,
                                  0.0,  0.0,  1.0,  0.0,
                                  0.0, -1.0,  0.0,  0.0,
                                  0.0,  0.0,  0.0,  1.0 };
// Angles (0, 90, 31)
const double ObliqueMatrix[] =  { 0.0, -0.515038, 0.857167, 0.0,
                                  0.0,  0.857167, 0.515038, 0.0,
                                 -1.0,  0.0,      0.0,      0.0,
                                  0.0,  0.0,      0.0,      1.0 };

class ResliceRender;

// Class to handle key press events.
class KeyCallback : public vtkCommand
{
public:
    static KeyCallback* New()
    {
        return new KeyCallback();
    }

    void Execute(vtkObject* caller, unsigned long eventId, void *calldata);
    void SetCallbackData(ResliceRender* reslice);

protected:
    ResliceRender* _reslice;
};

class ResliceRender
{
public:
    typedef enum _ORIENTATION
    {
        AXIAL = 0,
```

```
        SAGITTAL = 1,
        CORONAL = 2,
        OBLIQUE = 3
    } ORIENTATION;

    ResliceRender()
    {
        _orientation=AXIAL;
    }

    ~ResliceRender()
    {
        _transform->Delete();
        _reader->Delete();
        _reslice->Delete();
        _interactor->Delete();
        _imageViewer->Delete();

        _sphere->Delete();
        _sphereMapper->Delete();
        _sphereActor->Delete();

        _plane->Delete();
        _cutter->Delete();
        _polyTransform->Delete();
        _ROIMapper->Delete();
        _ROIActor->Delete();

        _annotation->Delete();
    }

    void CreatePipeline(const char* fileName)
    {
        vtkProperty2D* props;

        //_reader=vtkXMLImageDataReader::New();
        //_reader->SetFileName(fileName);
        //_reader->Update();

        //_reader=qzDICOMImageReader::New();
        _reader=vtkGDCMImageReader::New();

      //vtkDirectory *d = vtkDirectory::New();
      //d->Open(fileName);
      //d->Print( std::cout );
      gdcm::Directory d;
      d.Load(fileName);
      gdcm::Directory::FilenamesType const &files
       = d.GetFilenames();

  gdcm::IPPSorter s;
  s.SetComputeZSpacing( true );
  s.SetZSpacingTolerance( 1e-3 );
  bool b = s.Sort( files );
  if( !b )
    {
    std::cerr << "Failed to sort:" << fileName << std::endl;
    //return ;
    }
  //std::cout << "Sorting succeeded:" << std::endl;
  //s.Print( std::cout );

  //std::cout << "Found z-spacing:" << std::endl;
  //std::cout << s.GetZSpacing() << std::endl;
  double ippzspacing = s.GetZSpacing();

  const std::vector<std::string> & sorted = s.GetFilenames();
  vtkStringArray *vtkfiles = vtkStringArray::New();
  std::vector< std::string >::const_iterator it = sorted.begin();
  for( ; it != sorted.end(); ++it)
    {
    const std::string &f = *it;
    vtkfiles->InsertNextValue( f.c_str() );
    }

        //_reader->SetDirectoryName(fileName);
        //_reader->SetFileNames( d->GetFiles() );
        _reader->SetFileNames( vtkfiles );
        _reader->Update();

  const vtkFloatingPointType *spacing = _reader->GetOutput()->GetSpacing();
```

```
vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
v16->SetInput( _reader->GetOutput() );
v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
v16->Update();


      _threshold=vtkImageThreshold::New();
      _threshold->ThresholdByUpper(-3024.0);
      _threshold->ReplaceOutOn();
      _threshold->SetOutValue(0.0);
      _threshold->SetInputConnection(v16->GetOutputPort());

      _shift=vtkImageShiftScale::New();
      _shift->SetShift(0);
      _shift->SetScale(1);
      _shift->SetInputConnection(_threshold->GetOutputPort());

      // Initialize the reslice with an axial orientation.
      vtkSmartPointer<vtkMatrix4x4> matrix =
          vtkSmartPointer<vtkMatrix4x4>::New();
      matrix->Identity();

      _transform = vtkTransform::New();
      _transform->SetMatrix(matrix);

      _reslice = vtkImageReslice::New();
      _reslice->SetOutputDimensionality(3);

      // PROBLEM:
      // The original intent was to connect the same transform
      // to the vtkImageReslice and vtkTransformPolyDataFilter,
      // but the resulting reslices appear different using the
      // vtkTransform as opposed to explicitly setting the
      // reslice axes via SetResliceAxes.  Also, if the vtkTransform
      // is connected and orientated other than axial, the extents
      // don't seem to update resulting in VTK believing the slice
      // is out of range.

      //_reslice->SetResliceTransform(_transform);
      _reslice->SetResliceAxes(matrix);
      //_reslice->SetInputConnection(_reader->GetOutputPort());
      _reslice->SetInputConnection(_shift->GetOutputPort());

      // Create the sphere target shape.
      _sphere=vtkSphereSource::New();
      _sphere->SetRadius(7.0);
      _sphere->SetThetaResolution(16);
      _sphere->SetPhiResolution(16);
      _sphere->SetCenter(sphereCenter[0], sphereCenter[1], sphereCenter[2]);

      _sphereMapper=vtkPolyDataMapper::New();
      _sphereMapper->SetInputConnection(_sphere->GetOutputPort());

      _sphereActor=vtkActor::New();
      _sphereActor->SetMapper(_sphereMapper);
      _sphereActor->PickableOff();
      _sphereActor->GetProperty()->SetColor(1.0, 0.0, 0.0);
      _sphereActor->GetProperty()->SetEdgeColor(1.0, 0.0, 0.0);
      _sphereActor->GetProperty()->SetDiffuseColor(1.0, 0.0, 0.0);
      _sphereActor->SetVisibility(true);

      // Create the cutting pipeline.
      // This plane will be positioned in the original image coordinate
     system.
      _plane = vtkPlane::New();
      _plane->SetNormal(0.0, 0.0, 1.0);

      _cutter = vtkCutter::New();
      _cutter->SetInputConnection(_sphere->GetOutputPort());
      _cutter->SetCutFunction(_plane);
      _cutter->GenerateCutScalarsOn();
      _cutter->SetValue(0, 0.5);

      // The transform attached to _polyTransform should move the cut
      // ROI into the resliced coordinate system, which should be the
      // same as the coordinate system of the resliced images.
      // PROBLEM:  It doesn't.
      _polyTransform = vtkTransformPolyDataFilter::New();
      _polyTransform->SetTransform(_transform);
      _polyTransform->SetInputConnection(_cutter->GetOutputPort());
```

```
        _ROIMapper = vtkPolyDataMapper2D::New();
        _ROIMapper->SetInputConnection(_polyTransform->GetOutputPort());

    vtkCoordinate* coordinate = vtkCoordinate::New();
    coordinate->SetCoordinateSystemToWorld();
    _ROIMapper->SetTransformCoordinate(coordinate);

        _ROIActor = vtkActor2D::New();
        _ROIActor->SetMapper(_ROIMapper);

        // Make sure the cut can be seen, especially the edges.
        props=_ROIActor->GetProperty();
        props->SetLineWidth(2);
        props->SetOpacity(1.0);
//        props->EdgeVisibilityOn();
//        props->SetDiffuse(0.8);
//        props->SetSpecular(0.3);
//        props->SetSpecularPower(20);
//        props->SetRepresentationToSurface();
//        props->SetDiffuseColor(1.0, 0.0, 0.0);
//        props->SetEdgeColor(1.0, 0.0, 0.0);
        props->SetColor(1.0, 0.0, 0.0);

        _interactor = vtkRenderWindowInteractor::New();

        // Create the image viewer and add the actor with the cut ROI.
        _imageViewer = vtkImageViewer2::New();
        _imageViewer->SetupInteractor(_interactor);
        _imageViewer->SetSize(400, 400);
        _imageViewer->SetColorWindow(1024);
        _imageViewer->SetColorLevel(800);
        _imageViewer->SetInputConnection(_reslice->GetOutputPort());
        _imageViewer->GetImageActor()->SetOpacity(0.5);

        _annotation = vtkTextActor::New();
        _annotation->SetTextScaleModeToViewport();
        _imageViewer->GetRenderer()->AddActor(_annotation);

        // Add the cut shape actor to the renderer.
        _imageViewer->GetRenderer()->AddActor(_ROIActor);

        // Set up the key handler.
        vtkSmartPointer<KeyCallback> callback =
      vtkSmartPointer<KeyCallback>::New();
        callback->SetCallbackData(this);
        _interactor->AddObserver(vtkCommand::KeyPressEvent, callback);

        _interactor->Initialize();
    }

    void Start()
    {
        _interactor->Start();
    }

    void ResetOrientation()
    {
        vtkSmartPointer<vtkMatrix4x4> matrix =
            vtkSmartPointer<vtkMatrix4x4>::New();
        matrix->Identity();

        SetOrientation(matrix);
    }

    // Make sure the orientation of the vtkImageReslice and
    // vtkTransform are in sync.
    void SetOrientation(vtkMatrix4x4* matrix)
    {
        _reslice->SetResliceAxes(matrix);
        _reslice->Update();

    vtkMatrix4x4* inverse = vtkMatrix4x4::New();
    vtkMatrix4x4::Invert(matrix, inverse);

        _transform->SetMatrix(inverse);
        _transform->Update();
    }

    // Set the current slice of the current view.
    void SetSlice(int slice)
```

```
{
    std::strstream posString;

    double    center[3];
    double    spacing[3];
    double    origin[3];
    double    point[4];
    double    newPoint[4];

    vtkImageData* imageData;
    int newSlice;

    // Try to make sure the extents of the reslice are updated.
    // PROBLEM:  It doesn't seem to work when changing the orientation.
    imageData=vtkImageData::SafeDownCast(_reslice->GetOutput());
    imageData->UpdateInformation();

    // Let vtkImageViewer2 handle the slice limits.
    _imageViewer->SetSlice(slice);
    newSlice=GetSlice();

    imageData->GetCenter(center);
    imageData->GetSpacing(spacing);
    imageData->GetOrigin(origin);

    // Compute the position of the center of the slice based on the
    // spacing of the slices.  The resliced axis will always
    // be the "Z" axis.
    point[0]=center[0];
    point[1]=center[1];
    point[2]=(newSlice * spacing[2]) + origin[2];
    point[3]=1.0;

    // Convert the coordinate from the reslice coordinate system to the
    // original image coordinate system.
    // PROBLEM:  Logically this seems like it should have been multiplied
    // by the inverse to translate from the resliced coordinate system to
    // the original coordinate system.  However, multiplying by the inverse
    // sticks the plane in the wrong place completely.  Using the original
    // matrix at least gets the Z coordinate right.
    vtkMatrix4x4* matrix=_reslice->GetResliceAxes();
    vtkSmartPointer<vtkMatrix4x4> inverse =
        vtkSmartPointer<vtkMatrix4x4>::New();
    vtkMatrix4x4::Invert(matrix, inverse);

    matrix->MultiplyPoint(point, newPoint);
    _plane->SetOrigin(newPoint[0], newPoint[1], newPoint[2]);

    // Annotate the image.
    posString << "Position: (" << newPoint[0] << ", " << newPoint[1]
              << ", " << newPoint[2] << ")  Slice: " << newSlice;
    _annotation->SetInput(posString.str());

    _imageViewer->Render();
}

int GetSlice()
{
    return _imageViewer->GetSlice();
}

// Set the orientation of the view.
void SetOrientation(ResliceRender::ORIENTATION orientation)
{
    vtkCamera* camera=_imageViewer->GetRenderer()->GetActiveCamera();

    double spacing[3];
    double origin[3];
    double point[4];
    double newPoint[4];
    double initialPosition;
    double xDirCosine[3];
    double yDirCosine[3];
    double zDirCosine[3];
    double normal[3];

    vtkImageData* imageData;

    vtkSmartPointer<vtkMatrix4x4> matrix =
        vtkSmartPointer<vtkMatrix4x4>::New();
```

```
    _orientation=orientation;

    // Reset ViewUp
    camera->SetViewUp(0.0, 1.0, 0.0);

    // Compute the cut plane position to the input coordinate system.
    imageData=vtkImageData::SafeDownCast(_reslice->GetInput());
    imageData->UpdateInformation();
    imageData->GetSpacing(spacing);
    imageData->GetOrigin(origin);

    point[0]=origin[0];
    point[1]=origin[1];
    point[2]=origin[2];
    point[3]=1.0;

    switch (_orientation)
    {
    case AXIAL:
        matrix->DeepCopy(AxialMatrix);
        initialPosition=sphereCenter[2];
        break;

    case CORONAL:
        matrix->DeepCopy(CoronalMatrix);
        initialPosition=sphereCenter[1];
        break;

    case SAGITTAL:
        matrix->DeepCopy(SagittalMatrix);
        initialPosition=sphereCenter[0];
        break;

    case OBLIQUE:
        matrix->DeepCopy(ObliqueMatrix);
        initialPosition=sphereCenter[2];
        break;
    }

    // Move the origin from the original image coordinate system to the
    // resliced image coordinate system.
    matrix->MultiplyPoint(point, newPoint);
    matrix->SetElement(0, 3, newPoint[0]);
    matrix->SetElement(1, 3, newPoint[1]);
    matrix->SetElement(2, 3, newPoint[2]);

    ResetOrientation();
    SetOrientation(matrix);

    // Compute the cutting plane normal and set it.
    // PROBLEM:  If the transformation is connected rather than
    // using SetResliceAxes, the Direction Cosines do not reflect
    // the orientation of the vtkImageReslice.
    _reslice->GetResliceAxesDirectionCosines(xDirCosine, yDirCosine,
                                             zDirCosine);
    vtkMath::Cross(xDirCosine, yDirCosine, normal);
    _plane->SetNormal(normal);

    // Set the extents and spacing of the reslice to account for
    // all of the data.
    _reslice->SetOutputExtentToDefault();
    _reslice->SetOutputSpacing(spacing[0], spacing[0], spacing[0]);

    // Force the vtkImageViewer2 to update.
    // PROBLEM:  The whole extent does not seem to be set in time
    // for the first render.  This results in an error because the
    // slice is positioned outside the old bounds.
    _imageViewer->SetInput(NULL);
    _imageViewer->SetInputConnection(_reslice->GetOutputPort());

    _imageViewer->GetRenderer()->ResetCameraClippingRange();
    _imageViewer->GetRenderer()->ResetCamera();

    // Set the initial slice to be at the center of the sphere.
    // Divide by the spacing because this will be undone in SetSlice.
    SetSlice(initialPosition / spacing[0]);
}

vtkRenderWindowInteractor* GetInteractor()
{
    return _interactor;
```

```
    }

protected:
    ORIENTATION                 _orientation;

    //qzDICOMImageReader*        _reader;
    vtkGDCMImageReader*        _reader;
    vtkImageThreshold*          _threshold;
    vtkImageShiftScale*         _shift;
    vtkImageReslice*            _reslice;
    vtkRenderWindowInteractor*  _interactor;
    vtkImageViewer2*            _imageViewer;

    vtkSphereSource*            _sphere;
    vtkPolyDataMapper*          _sphereMapper;
    vtkActor*                   _sphereActor;

    vtkPlane*                   _plane;
    vtkCutter*                  _cutter;
    vtkTransform*               _transform;
    vtkTransformPolyDataFilter* _polyTransform;
    vtkPolyDataMapper2D*          _ROIMapper;
    vtkActor2D*                   _ROIActor;

    vtkTextActor*               _annotation;
};


// Catch KeyPress events.
// Up Arrow   - increases the slice
// Down Arrow - decreases the slice
// 'A'        - sets the view to Axial
// 'S'        - sets the view to Sagittal
// 'C'        - sets the view to Coronal
// 'O'        - set the view to Oblique

void KeyCallback::Execute(vtkObject* caller, unsigned long eventId, void *
      calldata)
{
    std::string sym=_reslice->GetInteractor()->GetKeySym();

    if (!sym.compare("Up"))
    {
        _reslice->SetSlice(_reslice->GetSlice() + 1);
    }
    else if (!sym.compare("Down"))
    {
        _reslice->SetSlice(_reslice->GetSlice() - 1);
    }
    else if ((!sym.compare("A")) || (!sym.compare("a")))
    {
        _reslice->SetOrientation(ResliceRender::AXIAL);
    }
    else if ((!sym.compare("C")) || (!sym.compare("c")))
    {
        _reslice->SetOrientation(ResliceRender::CORONAL);
    }
    else if ((!sym.compare("S")) || (!sym.compare("s")))
    {
        _reslice->SetOrientation(ResliceRender::SAGITTAL);
    }
    else if ((!sym.compare("O")) || (!sym.compare("o")))
    {
        _reslice->SetOrientation(ResliceRender::OBLIQUE);
    }
}

void KeyCallback::SetCallbackData(ResliceRender* reslice)
{
    _reslice=reslice;
}

// Usage:  ResliceSphere [fileName]
int main(int argc, char *argv[])
{
    ResliceRender render;

    if (argc == 1)
      {
      const char *root = gdcm::Testing::GetDataExtraRoot
      ();
```

```
      std::string dir3 = root;
      dir3 += "/gdcmSampleData/ForSeriesTesting/Dentist/images/";
      render.CreatePipeline(dir3.c_str());
      }
    else
      {
      render.CreatePipeline(argv[1]);
      }

    render.SetOrientation(ResliceRender::AXIAL);
    render.Start();

    return EXIT_SUCCESS;
}
```

27.116 ReWriteSCAsMR.py

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

"""
GDCM 1.x would write out MR Image Storage as Secondary Capture Object while
       still setting Rescale Slope/Intercept
and saving the Pixel Spacing in (0028,0030)
"""

import gdcm
import sys,os

def CheckSecondaryCaptureObjectIsMRImageStorage(r):
  ds = r.GetFile().GetDataSet()
  # Check Source Image Sequence
  if ds.FindDataElement( gdcm.Tag(0x0008,0x2112) ):
    sis = ds.GetDataElement( gdcm.Tag(0x0008,0x2112) )
    sqsis = sis.GetSequenceOfItems()
    if sqsis.GetNumberOfItems():
      item1 = sqsis.GetItem(1)
      nestedds = item1.GetNestedDataSet()
      if nestedds.FindDataElement( gdcm.Tag(0x0008,0x1150) ):
        ReferencedSOPClassUID = nestedds.GetDataElement( gdcm.Tag(0
      x0008,0x1150) )
        raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
        uids = gdcm.UIDs()
        # what is the actual object we are looking at ?
        ms = gdcm.MediaStorage()
        ms.SetFromDataSet(ds)
        msuid = ms.GetString()
        uids.SetFromUID( msuid )
        msuidname = uids.GetName() # real Media Storage Name
        uids.SetFromUID( raw )
        sqmsuidname = uids.GetName() # Source Image Sequence Media Storage Name
        # If object is SC and Source derivation is MRImageStorage then we can
       assume 'Pixel Spacing' is correct
        if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary
       Capture Image Storage' ):
          return True
  # in all other case simply return the currentspacing:
  return False

if __name__ == "__main__":
  r = gdcm.ImageReader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read():
    sys.exit(1)
  f = r.GetFile()
```

```
  if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
    # Special handling of the spacing:
    # GDCM 1.2.0 would not rewrite correcly DICOM Object and would always set
       them as 'Secondary Capture Image Storage'
    # while we would rather have 'MR Image Storage'
    gdcm.ImageHelper.SetForcePixelSpacing(
       True )
    mrspacing = gdcm.ImageHelper.GetSpacingValue
      ( r.GetFile() )
    # TODO: I cannot do simply the following:
    #image.SetSpacing( mrspacing )
    image.SetSpacing(0, mrspacing[0] )
    image.SetSpacing(1, mrspacing[1] )
    image.SetSpacing(2, mrspacing[2] )
    gdcm.ImageHelper.SetForceRescaleInterceptSlope
      ( True )
    ris = gdcm.ImageHelper.GetRescaleInterceptSlopeValue
      ( r.GetFile() )
    image.SetIntercept( ris[0] )
    image.SetSlope( ris[1] )

  outfilename = sys.argv[2]
  w = gdcm.ImageWriter()
  w.SetFileName( outfilename )
  w.SetFile( r.GetFile() )
  w.SetImage( image )
  if not w.Write():
    sys.exit(1)

  sys.exit(0)
```

27.117 rle2img.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to rewrite a ELSCINT1/PMSCT_RLE1 compressed
 * image so that it is readable by most 3rd party software (DICOM does
 * not specify this particular encoding).
 * This is required for the sake of interoperability with any standard
 * conforming DICOM system.
 *
 * Everything done in this code is for the sole purpose of writing
       interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Mauro Maiorca for bringing to our attention on this new ELSCINT1
 * compression algorithm : PMSCT_RLE1 (different from the 'LOSSLESS RICE')
 * See post at:
 * http://groups.google.com/group/comp.protocols.dicom/msg/f2b99bf706a7f8ca
 *
 * Thanks to Jesus Spinola, for more datasets,
 * http://www.itk.org/pipermail/insight-users/2008-April/025571.html
 *
 * And last but not least, a very big thank to Ivo van Poorten, without
 * whom we would still be looking at this compressed byte stream as if
 * it was RLE compressed.
 */
```

```
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

/* FIXME: Why is PhilipsLosslessRice.dcm a 512x512 image ... */
void delta_decode(const char *inbuffer, size_t length, std::vector<unsigned
       short> &output)
{
  // RLE pass
  std::vector<char> temp;
  for(size_t i = 0; i < length; ++i)
    {
    if( inbuffer[i] == (char)0xa5 )
      {
      //unsigned char repeat = (unsigned char)inbuffer[i+1] + 1;
      //assert( (unsigned char)inbuffer[i+1] != 255 );
      int repeat = (unsigned char)inbuffer[i+1] + 1;
      char value = inbuffer[i+2];
      while(repeat)
        {
        temp.push_back( value );
        --repeat;
        }
      i+=2;
      }
    else
      {
      temp.push_back( inbuffer[i] );
      }
    }

  // Delta encoding pass
  unsigned short delta = 0;
  for(size_t i = 0; i < temp.size(); ++i)
    {
    if( temp[i] == 0x5a )
      {
      unsigned char v1 = (unsigned char)temp[i+1];
      unsigned char v2 = (unsigned char)temp[i+2];
      int value = v2 * 256 + v1;
      output.push_back( value );
      delta = value;
      i+=2;
      }
    else
      {
      int value = temp[i] + delta;
      output.push_back( value );
      delta = value;
      }
    //assert( output[output.size()-1] == ref[output.size()-1] );
    }

  if ( output.size() % 2 )
    {
    output.resize( output.size() - 1 );
    }
  std::cout << length << " -> " << output.size() * 2 << std::endl;
}

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet
      ();

  // (07a1,1011) CS [PMSCT_RLE1]                                      # 10,1
       Tamar Compression Type
  const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"
      ELSCINT1");
  if( !ds.FindDataElement( tcompressiontype ) ) return 1;
  const gdcm::DataElement& compressiontype = ds.GetDataElement
```

```
    ( tcompressiontype );
  if ( compressiontype.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = compressiontype.GetByteValue
      ();
  std::string comprle = "PMSCT_RLE1";
  std::string comprgb = "PMSCT_RGB1";
  bool isrle = false;
  bool isrgb = false;
  if( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) ==
       0 )
    {
    isrle = true;
    }
  if( strncmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) ==
       0 )
    {
    isrgb = true;
    return 1;
    }
  if( !isrgb && !isrle ) return 1;

  const gdcm::PrivateTag tcompressedpixeldata(0x07a1,0x000a,"
      ELSCINT1");
  if( !ds.FindDataElement( tcompressedpixeldata) ) return 1;
  const gdcm::DataElement& compressionpixeldata = ds.
      GetDataElement( tcompressedpixeldata);
  if ( compressionpixeldata.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv2 = compressionpixeldata.
      GetByteValue();

  std::vector<unsigned short> buffer;
  delta_decode(bv2->GetPointer(), bv2->GetLength(), buffer);

  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0
      x0010) );
  pixeldata.SetVR( gdcm::VR::OW );
  pixeldata.SetByteValue( (char*)&buffer[0], buffer.size() * sizeof( unsigned
      short ) );
  // TODO we should check that decompress byte buffer match the expected size
       (row*col*...)

  // Add the pixel data element
  reader.GetFile().GetDataSet().Replace( pixeldata );

  reader.GetFile().GetHeader().SetDataSetTransferSyntax
      (
    gdcm::TransferSyntax::ExplicitVRLittleEndian
      );
  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );

  // Cleanup stuff:
  // remove the compressed pixel data:
  // FIXME: should I remove more private tags ? all of them ?
  // oh well this is just an example
  // use gdcm::Anonymizer::RemovePrivateTags if needed...
  writer.GetFile().GetDataSet().Remove(
      compressionpixeldata.GetTag() );
  std::string outfilename;
  if (argc > 2)
     outfilename = argv[2];
  else
     outfilename = "outrle.dcm";
  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr << "Failed to write" << std::endl;
    return 1;
    }

  std::cout << "success !" << std::endl;

  return 0;
}
```

27.118 rtstructapp.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMPolyDataReader.h"
#include "vtkGDCMPolyDataWriter.h"

#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkAppendPolyData.h"
#include "vtkImageData.h"

/*
 * Small example to read in a RTSTUCT and write it out (displays it too).
 */

// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm\n";
    return 1;
    }
  const char * filename = argv[1];
  const char * outfilename = argv[2];
  vtkGDCMPolyDataReader * reader =
      vtkGDCMPolyDataReader::New();
  reader->SetFileName( filename );
  reader->Update();

  //std::cout << reader->GetMedicalImageProperties()->GetStudyDate() <<
       std::endl;

  vtkGDCMPolyDataWriter * writer =
      vtkGDCMPolyDataWriter::New();
  writer->SetNumberOfInputPorts( reader->
      GetNumberOfOutputPorts() );
  writer->SetFileName( outfilename );
  for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
    writer->SetInput( num, reader->GetOutput(num) );
  //doesn't look like the medical properties are actually written out
  writer->SetMedicalImageProperties( reader->
      GetMedicalImageProperties() );
  writer->SetRTStructSetProperties( reader->
      GetRTStructSetProperties() );
  writer->Write();

  // print reader output:
  reader->Print( std::cout );
  // print first output:
  reader->GetOutput()->Print( std::cout );

 vtkAppendPolyData *append = vtkAppendPolyData::New();

 int n = reader->GetNumberOfOutputPorts();
 for(int i = 0; i < n; ++i)
```

```
   {
   append->AddInput( reader->GetOutput(i) );
   }

  // Now we'll look at it.
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  cubeMapper->SetInput( append->GetOutput());
  cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();

  vtkRenderer *renderer = vtkRenderer::New();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  renWin->AddRenderer(renderer);

  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renWin);

  renderer->AddActor(cubeActor);
  renderer->ResetCamera();
  renderer->SetBackground(1,1,1);

  renWin->SetSize(300,300);

  renWin->Render();
  iren->Start();

  reader->Delete();
  append->Delete();
  cubeMapper->Delete();
  cubeActor->Delete();
  renderer->Delete();
  renWin->Delete();
  iren->Delete();
  writer->Delete();

  return 0;
}
```

27.119 ScanDirectory.cs

This is a C# example on how to use gdcm::Scanner

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ScanDirectory.exe /path/to/gdcmData/
 */
using System;
using gdcm;

public class ScanDirectory
{
  public static int Main(string[] args)
    {
    string directory = args[0];
    Tag t = new Tag(0x8,0x8);

    Directory d = new Directory();
    uint nfiles = d.Load( directory );
```

```
    if(nfiles == 0) return 1;
    //System.Console.WriteLine( "Files:\n" + d.toString() );

    //Scanner s = new Scanner();
    SmartPtrScan sscan = Scanner.New();
    Scanner s = sscan.__ref__();
    SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(s, "MySimple");
    s.AddTag( t );
    bool b = s.Scan( d.GetFilenames() );
    if(!b) return 1;

    System.Console.WriteLine( "Scan:\n" + s.toString() );

    System.Console.WriteLine( "success" );
    return 0;
    }
}
```

27.120 ScanDirectory.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

import gdcm.*;
import gdcm.Reader;
import gdcm.LookupTable;
import java.io.File;
import java.io.*;
import java.awt.image.*;
import javax.imageio.ImageIO;

public class ScanDirectory
{
  public static class MyWatcher extends SimpleSubjectWatcher
    {
    public MyWatcher(Subject s) { super(s,"Override String"); }
    protected void ShowProgress(Subject caller, Event evt)
      {
      ProgressEvent pe = ProgressEvent.Cast(evt);
      System.out.println( "This is my progress: " + pe.GetProgress() );
      }
    }

  public static byte[] GetAsByte(Bitmap input)
    {
    long len = input.GetBufferLength();
    byte[] buffer = new byte[ (int)len ];
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
      {
      ImageChangePhotometricInterpretation icpi = new
      ImageChangePhotometricInterpretation();
      icpi.SetInput( input );
      icpi.SetPhotometricInterpretation(
        new PhotometricInterpretation(
          PhotometricInterpretation.PIType.MONOCHROME2 ) );
      if( icpi.Change() )
        {
        Bitmap output = icpi.GetOutput();
        output.GetArray( buffer );
        }
      return buffer;
      }
    else
      {
      input.GetArray( buffer );
```

```
      return buffer;
      }
    }
  public static short[] GetAsShort(Bitmap input)
    {
    long len = input.GetBufferLength(); // length in bytes
    short[] buffer = new short[ (int)len / 2 ];
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
      {
      ImageChangePhotometricInterpretation icpi = new
      ImageChangePhotometricInterpretation();
      icpi.SetInput( input );
      icpi.SetPhotometricInterpretation(
        new PhotometricInterpretation(
          PhotometricInterpretation.PIType.MONOCHROME2 ) );
      if( icpi.Change() )
        {
        Bitmap output = icpi.GetOutput();
        output.GetArray( buffer );
        }
      return buffer;
      }
    else
      {
      input.GetArray( buffer );
      return buffer;
      }
    }
  public static boolean WritePNG(Bitmap input, String outfilename )
    {
    int imageType = BufferedImage.TYPE_CUSTOM;
    PixelFormat pf = input.GetPixelFormat();
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    // We need to handle both public and private icon
    // It could well be that we are getting an RGB Icon or 16 bits Icon:
    ColorModel colorModel = null;
    if( pf.GetSamplesPerPixel() == 1 )
      {
      if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1
         || pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME2 )
        {
        if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
          {
          imageType = BufferedImage.TYPE_BYTE_GRAY;
          }
        else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT12 )
          {
          imageType = BufferedImage.TYPE_USHORT_GRAY;
          }
        else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT16 )
          {
          imageType = BufferedImage.TYPE_USHORT_GRAY;
          }
        }
      else if( pi.GetType() == PhotometricInterpretation.PIType.PALETTE_COLOR )
        {
        LookupTable lut = input.GetLUT();
        long rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
        byte[] rbuf = new byte[ (int)rl ];
        long rl2 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
        assert rl == rl2;
        long gl = lut.GetLUTLength( LookupTable.LookupTableType.GREEN );
        byte[] gbuf = new byte[ (int)gl ];
        long gl2 = lut.GetLUT( LookupTable.LookupTableType.GREEN, gbuf );
        assert gl == gl2;
        long bl = lut.GetLUTLength( LookupTable.LookupTableType.BLUE );
        byte[] bbuf = new byte[ (int)bl ];
        long bl2 = lut.GetLUT( LookupTable.LookupTableType.BLUE, bbuf );
        assert bl == bl2;
        colorModel = new IndexColorModel(8, (int)rl, rbuf, gbuf, bbuf);
        // For code below
        imageType = BufferedImage.TYPE_BYTE_GRAY;
        }
      }
    else if( pf.GetSamplesPerPixel() == 3 )
      {
      if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
        {
        // FIXME should be TYPE_3BYTE_RGB
        imageType = BufferedImage.TYPE_3BYTE_BGR;
```

```
        }
      }
    //System.out.println( "pf: " + pf.toString() );
    //System.out.println( "pi: " + pi.toString() );
    long width  = input.GetDimension(0);
    long height = input.GetDimension(0);
    BufferedImage bi;
    if( pi.GetType() == PhotometricInterpretation.PIType.PALETTE_COLOR )
      {
      bi = new BufferedImage(colorModel,
        colorModel.createCompatibleWritableRaster((int)width, (int)height),
        false, null);
      }
    else
      {
      bi = new BufferedImage((int)width,(int)height,imageType);
      }
    WritableRaster wr = bi.getRaster();
    //System.out.println( "imagetype: " + imageType );
    if( imageType == BufferedImage.TYPE_BYTE_GRAY
      || imageType == BufferedImage.TYPE_3BYTE_BGR )
      {
      byte[] buffer = GetAsByte( input );
      wr.setDataElements (0, 0, (int)width, (int)height, buffer);
      }
    else if( imageType == BufferedImage.TYPE_USHORT_GRAY )
      {
      short[] buffer = GetAsShort( input );
      wr.setDataElements (0, 0, (int)width, (int)height, buffer);
      }

    File outputfile = new File( outfilename );
    try {
      ImageIO.write(bi, "png", outputfile);
    } catch (IOException e) {
      return false;
    }
    return true;
    }

  public static void main(String[] args) throws Exception
    {
    String directory = args[0];

    Directory d = new Directory();
    long nfiles = d.Load( directory, true );
    if(nfiles == 0)
      {
      throw new Exception("No files found");
      }
//    System.out.println( "Files:\n" + d.toString() );
    FilenamesType fns = d.GetFilenames();

    //Scanner s = new Scanner();
    SmartPtrScan sscan = Scanner.New();
    Scanner s = sscan.__ref__();
    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(s, "MySimple");
    MyWatcher watcher = new MyWatcher(s);
    Tag[] tagarray = {
      new Tag(0x0010, 0x0010),    // PatientName
      new Tag(0x0010, 0x0020),    // PatientID
      new Tag(0x0010, 0x0030),    // PatientBirthDate
      new Tag(0x0010, 0x0040),    // PatientSex
      new Tag(0x0010, 0x1010),    // PatientAge
      new Tag(0x0020, 0x000d),    // StudyInstanceUID
      new Tag(0x0020, 0x0010),    // StudyID
      new Tag(0x0008, 0x0020),    // StudyDate
      new Tag(0x0008, 0x1030),    // StudyDescription
      new Tag(0x0020, 0x000e),    // SeriesInstanceUID
      new Tag(0x0020, 0x0011),    // SeriesNumber
      new Tag(0x0008, 0x0021),    // SeriesDate
      new Tag(0x0008, 0x103e),    // SeriesDescription
      new Tag(0x0008, 0x0090),    // ReferringPhysicianName
      new Tag(0x0008, 0x0060),    // Modality
      new Tag(0x0054, 0x0400),    // ImageID ?? Should be Instance number ??
      new Tag(0x0008, 0x0018),    // SOPInstanceUID
      new Tag(0x0008, 0x0032),    // AcquisitionTime
      new Tag(0x0008, 0x0033),    // ContentTime
      new Tag(0x0020, 0x0013),    // InstanceNumber
      new Tag(0x0020, 0x1041),    // SliceLocation
      new Tag(0x0018, 0x0050),    // SliceThickness ?? Eg. Enhanced MR Image
```

```
      Storage
    new Tag(0x0008, 0x0080),    // InstitutionName
    new Tag(0x0028, 0x1050),    // WindowCenter
    new Tag(0x0028, 0x1051),    // WindowWidth
  };
  for( Tag t : tagarray ) {
    //System.out.println( "Tag: " + t.toString() );
    s.AddTag( t );
  }
  boolean b = s.Scan( fns );
  if(!b)
    {
    throw new Exception("Could not scan");
    }

  for( long idx = 0; idx < fns.size(); ++idx )
    {
    Reader r = new Reader();
    String fn = fns.get( (int)idx );
    String outfn = fn + ".png";
    r.SetFileName( fn );
    TagSetType tst = new TagSetType();
    tst.insert( new Tag(0x7fe0,0x10) );
    b = r.ReadUpToTag( new Tag(0x88,0x200), tst );
    UIntArrayType dims = ImageHelper.GetDimensionsValue( r.GetFile() );
    if( b )
      {
      IconImageFilter iif = new IconImageFilter();
      System.out.println( "Processing: " + fn );

      iif.SetFile( r.GetFile() );
      b = iif.Extract();
      if( b )
        {
        Bitmap icon = iif.GetIconImage(0);
        WritePNG(icon, outfn);
        }
      else
        {
        ImageReader ir = new ImageReader();
        ir.SetFileName( fn );
        if( ir.Read() )
          {
          Image img = ir.GetImage();
          StringFilter sf = new StringFilter();
          sf.SetFile( r.GetFile() );
          String strval = sf.ToString( new Tag(0x0028,0x0120) );
          IconImageGenerator iig = new IconImageGenerator();
          iig.SetPixmap( img );
          iig.AutoPixelMinMax( true );
          try {
            double val = Double.parseDouble( strval );
            iig.SetOutsideValuePixel( val );
          }
          catch ( NumberFormatException e) {
            }
          iig.ConvertRGBToPaletteColor( false );
          long idims[] = { 128, 128};
          iig.SetOutputDimensions( idims );
          iig.Generate();
          Bitmap icon = iig.GetIconImage();
          WritePNG(icon, outfn);
          }
        }
      }
    }

  System.out.println( "Scan:\n" + s.toString() );

  System.out.println( "success" );
  }
}
```

27.121 ScanDirectory.py

```
##############################################################################
#
```

```
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

import gdcm
import sys,os

if __name__ == "__main__":
  directory = sys.argv[1]

  # Define the set of tags we are interested in
  t1 = gdcm.Tag(0x8,0x8);
  t2 = gdcm.Tag(0x10,0x10);

  # Iterate over directory
  d = gdcm.Directory();
  nfiles = d.Load( directory );
  if(nfiles == 0): sys.exit(1);
  # System.Console.WriteLine( "Files:\n" + d.toString() );

  filenames = d.GetFilenames()

  #  Get rid of any Warning while parsing the DICOM files
  gdcm.Trace.WarningOff()

  # instanciate Scanner:
  s = gdcm.Scanner();
  s.AddTag( t1 );
  s.AddTag( t2 );
  b = s.Scan( filenames );
  if(not b): sys.exit(1);

  print "success" ;
  #print s

  pttv = gdcm.PythonTagToValue( s.GetMapping( filenames[1] ) )
  pttv.Start()
  # iterate until the end:
  while( not pttv.IsAtEnd() ):
    # get current value for tag and associated value:
    # if tag was not found, then it was simply not added to the internal
       std::map
    # Warning value can be None
    tag = pttv.GetCurrentTag()
    value = pttv.GetCurrentValue()
    print tag,"->",value
    # increment iterator
    pttv.Next()

  sys.exit(0)
```

27.122 SendFileSCU.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
```

```
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm-gcc/bin
 * $ mono bin/SendFileSCU.exe server port input.dcm
 */
using System;
using gdcm;

public class SendFileSCU
{
  public static int Main(string[] args)
    {
    string server = args[0];
    ushort port = ushort.Parse(args[1]);
    string filename = args[2];

    bool b = CompositeNetworkFunctions.CEcho( server, port );
    if( !b ) return 1;

    FilenamesType files = new FilenamesType();
    files.Add( filename );
    b = CompositeNetworkFunctions.CStore( server, port, files );
    if( !b ) return 1;

    return 0;
    }
}
```

27.123 SimplePrint.cs

This is a C# example on how to use gdcm::SWIGDataSet

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
  Convertor convertor = new Convertor();
  int a = convertor.Convert<int>( some_int_blob );
  double b = convertor.Convert<double>( some_double_blob );
*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrint.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class SimplePrint
{
  public static void RecurseDataSet(File f, DataSet ds, string indent)
    {
    CSharpDataSet cds = new CSharpDataSet(ds);
    while(!cds.IsAtEnd())
      {
      DataElement de = cds.GetCurrent();
      // Compute VR from the toplevel file, and the currently processed
       dataset:
      VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );

      if( vr.Compatible( new VR(VR.VRType.SQ) ) )
        {
        uint uvl = (uint)de.GetVL(); // Test cast is ok
        System.Console.WriteLine( indent + de.GetTag().toString() + ":" + uvl )
      ; // why not ?
        //SequenceOfItems sq = de.GetSequenceOfItems();
        // GetValueAsSQ handle more cases than GetSequenceOfItems
```

```
        SmartPtrSQ sq = de.GetValueAsSQ();
        uint n = sq.GetNumberOfItems();
        for( uint i = 1; i <= n; i++) // item starts at 1, not 0
          {
          Item item = sq.GetItem( i );
          DataSet nested = item.GetNestedDataSet();
          RecurseDataSet( f, nested, indent + "  " );
          }
        }
      else
        {
        System.Console.WriteLine( indent + de.toString() );
        }
      cds.Next();
      }
    }

  public static int Main(string[] args)
    {
    string filename = args[0];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();

    RecurseDataSet( f, ds, "" );

    return 0;
    }
}
```

27.124 SimplePrintPatientName.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrintPatientName.exe gdcmData/012345.002.050.dcm
 */
/*
 This example was provided by Jonathan Morra /jonmorra gmail com/
 on the gdcm mailing list (Fri, 28 May 2010)
*/
using System;
using gdcm;

namespace GDCMTest
{
  class SimplePrintPatientName
    {
    static int Main(string[] args)
      {
      if (args.Length != 1)
        {
        Console.WriteLine("This program prints the patient name of a dicom file
       with gdcm");
        Console.WriteLine("Usage: [input.dcm]");
        return 1;
        }
```

```
      gdcm.Reader reader = new gdcm.Reader();
      reader.SetFileName(args[0]);
      bool ret = reader.Read();
      //TagSetType tst = new TagSetType();
      //tst.Add( new Tag(0x7fe0,0x10) );
      //bool ret = reader.ReadUpToTag( new Tag(0x88,0x200), tst );
      if( !ret )
        {
        return 1;
        }

      gdcm.File file = reader.GetFile();

      gdcm.StringFilter filter = new gdcm.StringFilter();
      filter.SetFile(file);
      string value = filter.ToString(new gdcm.Tag(0x0010, 0x0010));

      Console.WriteLine("Patient Name: " + value);
      return 0;
      }
    }
}
```

27.125 SimpleScanner.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Simple example to show how to use Scanner API.
 * It exposes the three different cases:
 *  - DICOM Attribute is present and has a value
 *  - DICOM Attribute is present and has no value
 *  - DICOM Attribute is not present at all
 * It also shows the purpose of the function 'IsKey' to detect whether or
 * not the file has been read by the gdcm::Scanner. Technically most of the
       time
 * if a file is not a 'Key' this is because it is not a DICOM file. You need to
       use
 * gdcm::System::FileExists to decide whether or not the file actually exist on
       the disk.
 *
 * It was tested on this particular image:
 * ./SimpleScanner gdcmData/012345.002.050.dcm
 */

#include "gdcmScanner.h"

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char filename_invalid[] = "this is a file that may not exist on this
       disk.dcm";

  gdcm::Scanner s;

  const gdcm::Tag tag_array[] = {
    gdcm::Tag(0x8,0x50),
    gdcm::Tag(0x8,0x51),
    gdcm::Tag(0x8,0x60),
  };
  s.AddTag( tag_array[0] );
  s.AddTag( tag_array[1] );
```

```
  s.AddTag( tag_array[2] );

  gdcm::Directory::FilenamesType filenames;
  filenames.push_back( filename );
  filenames.push_back( filename_invalid );

  if( !s.Scan( filenames ) )
    {
    return 1;
    }

  //s.Print( std::cout );


  if( s.IsKey( filename ) )
    {
    std::cout << "INFO:" << filename << " is a proper Key for the Scanner (this
       is a DICOM file)" << std::endl;
    }

  if( !s.IsKey( filename_invalid ) )
    {
    std::cout << "INFO:" << filename_invalid << " is not a proper Key for the
       Scanner (this is either not a DICOM file or file does not exist)" << std::endl;
    }

  gdcm::Scanner::TagToValue const &ttv = s.GetMapping
      (filename);

  const gdcm::Tag *ptag = tag_array;
  for( ; ptag != tag_array + 3; ++ptag )
    {
    gdcm::Scanner::TagToValue::const_iterator it = ttv.find( *ptag );
    if( it != ttv.end() )
      {
      std::cout << *ptag << " was properly found in this file" << std::endl;
      // it contains a pair of value. the first one is the actual tag, so the
       following is always true:
      //  *ptag == it->first
      // The second part is the actual value (stored as RAW strings). You will
       have to reinterpret this string
      // if VR for *ptag is not VR::VRASCII !
      const char *value = it->second;
      if( *value )
        {
        std::cout << "  It has the value: " << value << std::endl;
        }
      else
        {
        std::cout << "  It has no value (empty)" << std::endl;
        }
      }
    else
      {
      std::cout << "Sorry " << *ptag << " could not be found in this file" <<
      std::endl;
      }
    }

  return 0;
}
```

27.126 SortImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
```

```
/*
 */
#include "gdcmSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"


bool mysort(gdcm::DataSet const & ds1, gdcm::DataSet
      const & ds2 )
{
  //gdcm::Attribute<0x0020,0x0013> at1; // Instance Number
  gdcm::Attribute<0x0018,0x1060> at1; // Trigger
       Time
  gdcm::Attribute<0x0020,0x0032> at11; // Image
       Position (Patient)
  at1.Set( ds1 );
  at11.Set( ds1 );
  //gdcm::Attribute<0x0020,0x0013> at2;
  gdcm::Attribute<0x0018,0x1060> at2;
  gdcm::Attribute<0x0020,0x0032> at22;
  at2.Set( ds2 );
  at22.Set( ds2 );
  if( at11 == at22 )
    {
    return at1 < at2;
    }
  return at11 < at22;
}

bool mysort_part1(gdcm::DataSet const & ds1, gdcm::DataSet
       const & ds2 )
{
  gdcm::Attribute<0x0018,0x1060> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0018,0x1060> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

bool mysort_part2(gdcm::DataSet const & ds1, gdcm::DataSet
       const & ds2 )
{
  gdcm::Attribute<0x0020,0x0032> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0032> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

// technically all files are in the same Frame of Reference, so this function
// should be a no-op
bool mysort_dummy(gdcm::DataSet const & ds1, gdcm::DataSet
       const & ds2 )
{
  gdcm::Attribute<0x0020,0x0052> at1; //
       FrameOfReferenceUID
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0052> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

int main(int argc, char *argv[])
{
  const char *dirname = argv[1];
  gdcm::Directory dir;
  unsigned int nfiles = dir.Load( dirname );

  dir.Print( std::cout );

  gdcm::Sorter sorter;
  sorter.SetSortFunction( mysort );
  sorter.Sort( dir.GetFilenames() );

  std::cout << "Sorter:" << std::endl;
  sorter.Print( std::cout );

  gdcm::Sorter sorter2;
  sorter2.SetSortFunction( mysort_part1 );
  sorter2.StableSort( dir.GetFilenames() );
```

```
  sorter2.SetSortFunction( mysort_part2 );
  sorter2.StableSort( sorter2.GetFilenames() ); //
       IMPORTANT
  sorter2.SetSortFunction( mysort_dummy );
  sorter2.StableSort( sorter2.GetFilenames() ); //
       IMPORTANT

  std::cout << "Sorter2:" << std::endl;
  sorter2.Print( std::cout );

  gdcm::Scanner s;
  s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
  //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
  s.Scan( dir.GetFilenames() );

  //s.Print( std::cout );

  // Count how many different IPP there are:
  const gdcm::Scanner::ValuesType &values = s.
      GetValues();
  unsigned int nvalues = values.size();
  std::cout << "There are " << nvalues << " different type of values" <<
      std::endl;

  //std::cout << "nfiles=" << nfiles << std::endl;
  if( nfiles % nvalues != 0 )
    {
    std::cerr << "Impossible: this is a not a proper series" << std::endl;
    return 1;
    }
  std::cout << "Series is composed of " << (nfiles/nvalues) << " different 3D
       volumes" << std::endl;

  return 0;
}
```

27.127 SortImage.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################

"""
Usage:

 python SortImage.py dirname
"""

import gdcm
import sys

def PrintProgress(object, event):
  assert event == "ProgressEvent"
  print "Progress:", object.GetProgress()

def MySort(ds1, ds2):
  # compare ds1
  return False

if __name__ == "__main__":

  dirname = sys.argv[1]
  d = gdcm.Directory()
  d.Load( dirname )

  print d
```

```
  sorter = gdcm.Sorter()
  sorter.SetSortFunction( MySort )
  #sorter.AddObserver( "ProgressEvent", PrintProgress )
  sorter.Sort( d.GetFilenames() )

  print "Sorter:"
  print sorter
```

27.128 SortImage2.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/SortImage.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class SortImage2
{
  bool mysort(DataSet ds1, DataSet ds2)
    {
    return false;
    }

  public static int Main(string[] args)
    {
    Sorter sorter = new Sorter();
    sorter.SetSortFunction( mysort );

    return 0;
    }
}
```

27.129 StandardizeFiles.cs

This is a C++ example on how to use gdcm::ImageChangeTransferSyntax

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Simple C# example to show how one would 'Standardize' a DICOM File-Set
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/StandardizeFiles.exe input_path output_path
```

```
 */
using System;
using gdcm;

public class StandardizeFiles
{
  public static bool ProcessOneFile( string filename, string outfilename )
    {
    PixmapReader reader = new PixmapReader();
    reader.SetFileName( filename );
    if( !reader.Read() )
      {
      System.Console.WriteLine( "Could not read: " + filename );
      return false;
      }

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetForce( false ); // do we really want to recompress when input is
       alread compressed in same alg ?
    change.SetCompressIconImage( false ); // Keep it simple
    change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.
      JPEG2000Lossless ) );
    change.SetInput( reader.GetPixmap() );
    if( !change.Change() )
      {
      System.Console.WriteLine( "Could not change: " + filename );
      return false;
      }

    gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

    PixmapWriter writer = new PixmapWriter();
    writer.SetFileName( outfilename );
    writer.SetFile( reader.GetFile() );
    gdcm.Bitmap bitout = change.GetOutput();
    gdcm.Pixmap pixout = (gdcm.Pixmap)bitout;
    //System.Console.WriteLine( "Debug: " + pixout.toString() );

    writer.SetPixmap( pixout );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write: " + outfilename );
      return false;
      }

    return true;
    }

  public static int Main(string[] args)
    {
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Standardize
       App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot()
       );

    string dir1 = args[0];
    string dir2 = args[1];

    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
      {
      System.Console.WriteLine( "Input directory: " + dir1 + " does not exist.
       Sorry" );
      return 1;
      }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
      {
      System.Console.WriteLine( "Output directory: " + dir2 + " does not exist.
       Sorry" );
      return 1;
      }

    Directory d = new Directory();
```

```
    uint nfiles = d.Load( dir1, true );
    if(nfiles == 0) return 1;

    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
      {
      string filename = filenames[ (int)i ];
      string outfilename = filename.Replace( dir1, dir2 );
      System.Console.WriteLine( "Filename: " + filename );
      System.Console.WriteLine( "Out Filename: " + outfilename );
      if( !ProcessOneFile( filename, outfilename ) )
        {
        System.Console.WriteLine( "Could not process filename: " + filename );
        //return 1;
        }
      }


    return 0;
    }
}
```

27.130 StreamImageReaderTest.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcmStreamImageReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmSystem.h"
#include "gdcmFilename.h"
#include "gdcmByteSwap.h"
#include "gdcmTrace.h"
#include "gdcmTesting.h"
#include "gdcmImageHelper.h"
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmMediaStorage.h"
#include "gdcmRAWCodec.h"
#include "gdcmJPEGLSCodec.h"
#include "gdcmUIDGenerator.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

bool StreamImageRead(gdcm::StreamImageWriter &
      theStreamWriter,
  const char* filename, const char* outfilename, int resolution)
{
  gdcm::StreamImageReader reader;

  reader.SetFileName( filename );

  if (!reader.ReadImageInformation())
    {
    std::cerr << "unable to read image information" << std::endl;
    return 1; //unable to read tags as expected.
    }
  //let's be tricky; each image will be read in portions, first the top half,
      then the bottom
  //that way, we can test how the stream handles fragmentation of the data
  //we could also loop this to get various different size combinations, but I'm
      not sure
```

```
  //that's useful, yet.
  std::vector<unsigned int> extent =
    gdcm::ImageHelper::GetDimensionsValue(
      reader.GetFile());
  // std::cout << extent[0];
  //at this point, these values aren't used, but may be in the future
  //unsigned short xmin = 0;
  //unsigned short xmax = extent[0];
  //unsigned short ymin = 0;
  //unsigned short ymax = extent[1];
  //unsigned short zmin = 0;
  //unsigned short zmax = extent[2];

  std::cout<< "\n Row: "<<extent[0] <<"\n Col :"<< extent[1]<< "\n Resolution :
      "<< extent[2] << std::endl;

  int a =1;
  for (int i=1; i<=(extent[2]-resolution);++i)
    a = a*2;

  reader.DefinePixelExtent(0, extent[0]/a, 0, extent[1]/a,
      resolution-1, resolution);

  unsigned long len = reader.DefineProperBufferLength()
      ;
  char* finalBuffer = new char[len];
  memset(finalBuffer, 0, sizeof(char)*len);

  if (reader.CanReadImage())
    {
    bool result = reader.Read(finalBuffer, len);
    if( !result )
      {
      std::cout << "res2 failure:" << filename << std::endl;
      delete [] finalBuffer;
      return 1;
      }
    else
      {
      std::cout<< "Able to read";
      }
    }
  else
    {
    std::cerr<< "Not able to put in buffer"<< std::endl;
    }
/*
    //now, read in smaller buffer extents
    reader.DefinePixelExtent(xmin, xmax, ymin, ymax);
    len = reader.DefineProperBufferLength();

    char* buffer = new char[len];
    bool res2 = reader.Read(buffer, len);
    if( !res2 ){
      std::cerr << "res2 failure:" << filename << std::endl;
      return 1;
    }
    //copy the result into finalBuffer
    memcpy(finalBuffer, buffer, len);

    //now read the next half of the image
    ymin = ymax;
    ymax = extent[1];

    reader.DefinePixelExtent(xmin, xmax, ymin, ymax);

    //std::cerr << "Success to read image from file: " << filename <<
       std::endl;
    unsigned long len2 = reader.DefineProperBufferLength();

    char* buffer2 = new char[len2];
    bool res3 = reader.Read(buffer2, len2);
    if( !res3 ){
      std::cerr << "res3 failure:" << filename << std::endl;
      return 1;
    }
    //copy the result into finalBuffer
    memcpy(&(finalBuffer[len]), buffer2, len2);

    delete [] buffer;
    delete [] buffer2;
```

```
*/

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  file.GetHeader().SetDataSetTransferSyntax(
      gdcm::TransferSyntax::ExplicitVRLittleEndian
       );

  gdcm::UIDGenerator uid;
  gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP
       Instance UID
  de.SetVR( gdcm::VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );

  gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
  de1.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms(
      gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
       );
  de1.SetByteValue( ms.GetString(), strlen(ms.GetString
      ()));
  ds.Insert( de1 );

  const char mystr[] = "MONOCHROME2 ";
  gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  de2.SetVR( gdcm::VR::CS );
  de2.SetByteValue(mystr, strlen(mystr));
  ds.Insert( de2 );

   gdcm::Attribute<0x0028,0x0008>
      Number_Of_Frames = {1};
   ds.Insert( Number_Of_Frames.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0010> row = {extent[0
      ]/a};//
   ds.Insert( row.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0011> col = {extent[1
      ]/a};//
   ds.Insert( col.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0100> at = {8};
   ds.Insert( at.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0002> at1 = {1};//
   ds.Insert( at1.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0101> at2 = {8};
   ds.Insert( at2.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0102> at3 = {7};
   ds.Insert( at3.GetAsDataElement() );
    /*
   ds1.Remove( gdcm::Tag(0x0028,0x0008) );

   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
   ds1.Insert( Number_Of_Frames.GetAsDataElement() );
*/
   theStreamWriter.SetFile(file);

   if (!theStreamWriter.WriteImageInformation())
     {
     std::cerr << "unable to write image information" << std::endl;
     return 1; //the CanWrite function should prevent getting here, else,
     //that's a test failure
     }
    std::vector<unsigned int> extent1 = gdcm::ImageHelper::GetDimensionsValue
      (file);

    unsigned short xmax = extent1[0];
    unsigned short ymax = extent1[1];
    unsigned short theChunkSize = 1;
    unsigned short ychunk = extent1[1]/theChunkSize; //go in chunk sizes of
       theChunkSize
    unsigned short zmax = 1;
```

```
    std::cout<< "\n Row: "<<extent1[0] <<"\n Col :"<< extent1[1]<< "\n
       Resolution :"<< extent1[2] << std::endl;

    if (xmax == 0 || ymax == 0)
      {
      std::cerr << "Image has no size, unable to write zero-sized image." <<
      std::endl;
      return 0;
      }

    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure
       to grab
    //the bytes sequentially.  So, store how far you got in the buffer with
       each iteration.

    for (z = 0; z < zmax; ++z){
      for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z
      , z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength
      ();
        std::cout << "\n" <<len;
        char* finalBuffer1 = new char[len];
        memcpy(finalBuffer1, &(finalBuffer[prevLen]), len);
        std::cout << "\nable to write";

        if (!theStreamWriter.Write(finalBuffer1, len)){
          std::cerr << "writing failure:" << "output.dcm" << " at y = " << y <<
       " and z= " << z << std::endl;
          delete [] finalBuffer1;
          delete [] finalBuffer;
          return 1;
        }
        delete [] finalBuffer1;
        prevLen += len;
      }
    }
   delete [] finalBuffer;
   std::cout << "all is set";

  return true;
}


int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr << argv[0] << " input.dcm output.dcm Resolution" << std::endl;
    return 1;
    }

  const char *filename = argv[1];
  const char *outfilename = argv[2];
  char *res = argv[3];

  int resolution = atoi(res);

  gdcm::StreamImageWriter theStreamWriter;

  std::ofstream of;
  of.open( outfilename, std::ios::out | std::ios::binary );
  theStreamWriter.SetStream(of);

  // else
  // First of get rid of warning/debug message
  gdcm::Trace::DebugOn();
  gdcm::Trace::WarningOn();

  if(!StreamImageRead( theStreamWriter, filename, outfilename, resolution))
    return 1;

  uint16_t firstTag1 =  0xfffe;
  uint16_t secondTag1 = 0xe0dd;
  uint32_t thirdTag1 =  0x00000000;
  //uint16_t fourthTag1 = 0xffff;
  const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
  char* tmpBuffer2 = new char[theBufferSize1];
```

```
  memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
  memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
  memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
  //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
  assert( of && !of.eof() && of.good() );
  of.write(tmpBuffer2, theBufferSize1);
  of.flush();
  assert( of );

  return 0;
}
```

27.131 TestByteSwap.cxx

This is a C++ example on how to use gdcm::ByteSwap

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.h"

#include <string.h> // memcpy

int myfunc()
{
  char vl_str[4];
  const char raw[] = "\000\000\000\004";
  memcpy(vl_str, raw, 4);
  uint32_t vl;
  gdcm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem
      ((uint32_t*)(&vl_str), gdcm::SwapCode::BigEndian, 1);
  memcpy(&vl, vl_str, 4);
  if( vl != 0x00000004 )
    {
    std::cerr << std::hex << "vl: " << vl << std::endl;
    return 1;
    }

  gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem
      (vl, gdcm::SwapCode::LittleEndian);
  if( vl != 0x00000004 )
    {
    std::cerr << std::hex << "vl: " << vl << std::endl;
    return 1;
    }

  gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem
      (vl, gdcm::SwapCode::BigEndian);
  std::cout << std::hex << "vl: " << vl << std::endl;
  if( vl != 0x4000000 )
    {
    return 1;
    }

  return 0;
}

int TestByteSwap(int , char *[])
{
  gdcm::SwapCode sc = gdcm::SwapCode::Unknown
      ;
  if ( gdcm::ByteSwap<uint16_t>::SystemIsBigEndian
      () )
    {
```

```
  sc = gdcm::SwapCode::BigEndian;
  }
else if ( gdcm::ByteSwap<uint16_t>::SystemIsLittleEndian
    () )
  {
  sc = gdcm::SwapCode::LittleEndian;
  }
if( sc == gdcm::SwapCode::Unknown )
  {
  return 1;
  }

std::cout << "sc: " << sc << std::endl;

uint16_t t = 0x1234;
gdcm::ByteSwap<uint16_t>::SwapFromSwapCodeIntoSystem
    (t, sc);
if( sc == gdcm::SwapCode::BigEndian )
  {
  if( t != 0x3412 )
    {
    std::cerr << std::hex << "t: " << t << std::endl;
    return 1;
    }
  // ok test pass rest value to old one
  t = 0x1234;
  }
else if ( sc == gdcm::SwapCode::LittleEndian )
  {
  if( t != 0x1234 )
    {
    std::cerr << std::hex << "t: " << t << std::endl;
    return 1;
    }
  }

char n[2];
memcpy(n, &t, 2 );
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
    ((uint16_t*)n, sc, 1);
uint16_t tn = *((uint16_t*)n);
if( sc == gdcm::SwapCode::BigEndian )
  {
  if( tn != 0x3412 )
    {
    std::cerr << std::hex << "tn: " << tn << std::endl;
    return 1;
    }
  // ok test pass rest value to old one
  t = 0x1234;
  }
else if ( sc == gdcm::SwapCode::LittleEndian )
  {
  if( tn != 0x1234 )
    {
    std::cerr << std::hex << "tn: " << tn << std::endl;
    return 1;
    }
  }
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
    ((uint16_t*)n, gdcm::SwapCode::BigEndian, 1);
tn = *((uint16_t*)n);
if( sc == gdcm::SwapCode::LittleEndian )
  {
  if( tn != 0x3412 )
    {
    std::cerr << std::hex << "tn: " << tn << std::endl;
    return 1;
    }
  }
else if ( sc == gdcm::SwapCode::BigEndian )
  {
  if( tn != 0x1234 )
    {
    std::cerr << std::hex << "tn: " << tn << std::endl;
    return 1;
    }
  }

if( myfunc() )
  {
```

```
    return 1;
    }

  uint16_t array[] = { 0x1234 };
  gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
      (array,
    gdcm::SwapCode::BigEndian,2);
   if ( array[0] != 0x3412 )
     {
     return 1;
     }

  return 0;
}
```

27.132 TestReader.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmFile.h"
#include "gdcmTesting.h"
#include "gdcmMediaStorage.h"

int TestRead(const char* filename, bool verbose = false)
{
  if( verbose )
  std::cout << "TestRead: " << filename << std::endl;

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if ( !reader.Read() )
    {
    std::cerr << "TestReadError: Failed to read: " << filename << std::endl;
    return 1;
    }

//commenting out the fmi and ds to avoid warnings
  //const gdcm::FileMetaInformation &h = reader.GetFile().GetHeader();
  //std::cout << h << std::endl;

  //const gdcm::DataSet &ds = reader.GetFile().GetDataSet();
  //std::cout << ds << std::endl;

  const char *ref = gdcm::Testing::GetMediaStorageFromFile
      (filename);

  gdcm::MediaStorage ms;
  ms.SetFromFile( reader.GetFile() );
  if( ms.IsUndefined() && ref && *ref != 0 )
    {
    std::cerr << "TestReadError: MediaStorage: " << filename << std::endl;
    std::cerr << "It should be instead: " << ref << std::endl;
    return 1;
    }

  // Make sure it is the right one:

  if( ref && *ref != 0 && ms != gdcm::MediaStorage::GetMSType
      (ref) )
    {
    std::cerr << "Error: Found MediaStorage: " << ms << " for " << filename <<
      std::endl;
    std::cerr << "It should be instead: " << ref << std::endl;
    return 1;
```

```
    }

  return 0;
}

int TestReader(int argc, char *argv[])
{
  if( argc == 2 )
    {
    const char *filename = argv[1];
    return TestRead(filename, true);
    }

  // else
  gdcm::Trace::DebugOff();
  gdcm::Trace::WarningOff();
  int r = 0, i = 0;
  const char *filename;
  const char * const *filenames = gdcm::Testing::GetFileNames
      ();
  while( (filename = filenames[i]) )
    {
    r += TestRead( filename );
    ++i;
    }

  return r;
}
```

27.133 TestReader.py

This is a C++ example on how to use gdcm::Reader

```
##############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
##############################################################################

import gdcm
import os,sys

def TestRead(filename, verbose = False):
  r = gdcm.Reader()
  r.SetFileName( filename )
  sucess = r.Read()
  #if verbose: print r.GetFile()
  if verbose: print r.GetFile().GetDataSet()
  return sucess

if __name__ == "__main__":
  sucess = 0
  try:
    filename = os.sys.argv[1]
    sucess += TestRead( filename, True )
  except:
    # loop over all files:
    gdcm.Trace.DebugOff()
    gdcm.Trace.WarningOff()
    t = gdcm.Testing()
    nfiles = t.GetNumberOfFileNames()
    for i in range(0,nfiles):
      filename = t.GetFileName(i)
      sucess += TestRead( filename )


  # Test succeed ?
  sys.exit(sucess == 0)
```

27.134 threadgdcm.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmDirectory.h"
#include "gdcmSystem.h"

#include "vtkImageData.h"
#include "vtkStructuredPointsWriter.h"

#include <pthread.h>

struct threadparams
{
  const char **filenames;
  unsigned int nfiles;
  char *scalarpointer;
// TODO I should also pass in the dim of the reference image just in case
};

void *ReadFilesThread(void *voidparams)
{
  const threadparams *params = static_cast<const threadparams *> (voidparams);

  const unsigned int nfiles = params->nfiles;
  for(unsigned int file = 0; file < nfiles; ++file)
    {
    /*
    // TODO: update progress
    pthread_mutex_lock(&params->lock);
    //section critique
    ReadingProgress+=params->stepProgress;
    pthread_mutex_unlock(&params->lock);
    */
    const char *filename = params->filenames[file];
    //std::cerr << filename << std::endl;

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    try
      {
      if( !reader.Read() )
        {
        std::cerr << "Failed to read: " << filename << std::endl;
        break;
        }
      }
    catch( ... )
      {
      std::cerr << "Failed to read: " << filename << std::endl;
      break;
      }

    const gdcm::Image &image = reader.GetImage();
    unsigned long len = image.GetBufferLength();
    char * pointer = params->scalarpointer;
#if 0
    char *tempimage = new char[len];
    image.GetBuffer(tempimage);

    memcpy(pointer + file*len, tempimage, len);
    delete[] tempimage;
#else
    char *tempimage = pointer + file * len;
    image.GetBuffer(tempimage);
#endif
    }
```

```
  return voidparams;
}

void ShowFilenames(const threadparams &params)
{
  std::cout << "start" << std::endl;
  for(unsigned int i = 0; i < params.nfiles; ++i)
    {
    const char *filename = params.filenames[i];
    std::cout << filename << std::endl;
    }
  std::cout << "end" << std::endl;
}

void ReadFiles(unsigned int nfiles, const char *filenames[])
{
  // \precondition: nfiles > 0
  assert( nfiles > 0 );
  const char *reference= filenames[0]; // take the first image as reference

  gdcm::ImageReader reader;
  reader.SetFileName( reference );
  if( !reader.Read() )
    {
    // That would be very bad...
    assert(0);
    }

  const gdcm::Image &image = reader.GetImage();
  gdcm::PixelFormat pixeltype = image.GetPixelFormat
      ();
  unsigned long len = image.GetBufferLength();
  const unsigned int *dims = image.GetDimensions();
  unsigned short pixelsize = pixeltype.GetPixelSize();
  (void)pixelsize;
  assert( image.GetNumberOfDimensions() == 2 );

  vtkImageData *output = vtkImageData::New();
  output->SetDimensions(dims[0], dims[1], nfiles);

  switch( pixeltype )
    {
  case gdcm::PixelFormat::INT8:
#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION >
       5 )
    output->SetScalarType ( VTK_SIGNED_CHAR );
#else
    output->SetScalarType ( VTK_CHAR );
#endif
    break;
  case gdcm::PixelFormat::UINT8:
    output->SetScalarType ( VTK_UNSIGNED_CHAR );
    break;
  case gdcm::PixelFormat::INT16:
    output->SetScalarType ( VTK_SHORT );
    break;
  case gdcm::PixelFormat::UINT16:
    output->SetScalarType ( VTK_UNSIGNED_SHORT );
    break;
  case gdcm::PixelFormat::INT32:
    output->SetScalarType ( VTK_INT );
    break;
  case gdcm::PixelFormat::UINT32:
    output->SetScalarType ( VTK_UNSIGNED_INT );
    break;
  default:
    assert(0);
    }

  output->SetNumberOfScalarComponents ( pixeltype.GetSamplesPerPixel
      () );

  output->AllocateScalars();
  char * scalarpointer = static_cast<char*>(output->GetScalarPointer());

  const unsigned int nthreads = 4;
  threadparams params[nthreads];

  //pthread_mutex_t lock;
  //pthread_mutex_init(&lock, NULL);
```

```
  pthread_t *pthread = new pthread_t[nthreads];

  // There is nfiles, and nThreads
  assert( nfiles > nthreads );
  const unsigned int partition = nfiles / nthreads;
  for (unsigned int thread=0; thread < nthreads; ++thread)
    {
    params[thread].filenames = filenames + thread * partition;
    params[thread].nfiles = partition;
    if( thread == nthreads - 1 )
      {
      // There is slightly more files to process in this thread:
      params[thread].nfiles += nfiles % nthreads;
      }
    assert( thread * partition < nfiles );
    params[thread].scalarpointer = scalarpointer + thread * partition * len;
    //assert( params[thread].scalarpointer < scalarpointer + 2 * dims[0] *
       dims[1] * dims[2] );
    // start thread:
    int res = pthread_create( &pthread[thread], NULL, ReadFilesThread, &params[
      thread]);
    if( res )
      {
      std::cerr << "Unable to start a new thread, pthread returned: " << res <<
       std::endl;
      assert(0);
      }
    //ShowFilenames(params[thread]);
    }
// DEBUG
  unsigned int total = 0;
  for (unsigned int thread=0; thread < nthreads; ++thread)
    {
    total += params[thread].nfiles;
    }
  assert( total == nfiles );
// END DEBUG

  for (unsigned int thread=0;thread<nthreads;thread++)
    {
    pthread_join( pthread[thread], NULL);
    }
  delete[] pthread;

  //pthread_mutex_destroy(&lock);

  // For some reason writing down the file is painfully slow...
  vtkStructuredPointsWriter *writer = vtkStructuredPointsWriter::New();
  writer->SetInput( output );
  writer->SetFileName( "/tmp/threadgdcm.vtk" );
  writer->SetFileTypeToBinary();
  //writer->Write();
  writer->Delete();

  //output->Print( std::cout );
  output->Delete();
}

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr << argv[0] << " [directory|list of filenames]\n";
    return 1;
    }

  // Check if user pass in a single directory
  if( argc == 2 && gdcm::System::FileIsDirectory(
      argv[1] ) )
    {
    gdcm::Directory d;
    d.Load( argv[1] );
    gdcm::Directory::FilenamesType l = d.
      GetFilenames();
    const unsigned int nfiles = l.size();
    const char **filenames = new const char* [ nfiles ];
    for(unsigned int i = 0; i < nfiles; ++i)
      {
      filenames[i] = l[i].c_str();
      }
```

```
    ReadFiles(nfiles, filenames);
    delete[] filenames;
    }
  else
    {
    // Simply copy all filenames into the vector:
    const char **filenames = const_cast<const char**>(argv+1);
    const unsigned int nfiles = argc - 1;
    ReadFiles(nfiles, filenames);
    }


  return 0;
}
```

27.135 TraverseModules.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 */

#include "gdcmDefs.h"
#include "gdcmGlobal.h"
#include "gdcmIODs.h"
#include "gdcmIOD.h"
#include "gdcmMacros.h"
#include "gdcmIODEntry.h"
#include "gdcmModules.h"
#include "gdcmModule.h"
#include "gdcmAnonymizer.h"
#include "gdcmDicts.h"

int main(int , char *[])
{
  using namespace gdcm;
  static Global &g = Global::GetInstance();

  if( !g.LoadResourcesFiles() )
    {
    return 1;
    }

  static const Defs &defs = g.GetDefs();
  static const Modules &modules = defs.GetModules();
  static const IODs &iods = defs.GetIODs();
  static const Macros &macros = defs.GetMacros();
  static const Dicts &dicts = g.GetDicts();

  std::vector<Tag> tags =
      gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
      ();
  for( std::vector<Tag>::const_iterator tit = tags.begin(); tit != tags.end();
      ++tit )
    {
    const Tag &tag = *tit;
    const DictEntry &dictentry = dicts.GetDictEntry(tag);
    std::cout << "Processing Attribute: " << tag << " " << dictentry <<
      std::endl;

    IODs::IODMapTypeConstIterator it = iods.Begin(
      );
    for( ; it != iods.End(); ++it )
      {
      const IODs::IODName &name = it->first;
      const IOD &iod = it->second;
```

```
      const unsigned int niods = iod.GetNumberOfIODs();
      // Iterate over each iod entry in order:
      for(unsigned int idx = 0; idx < niods; ++idx)
        {
        const IODEntry &iodentry = iod.GetIODEntry(idx);
        const char *ref = iodentry.GetRef();
        //Usage::UsageType ut = iodentry.GetUsageType();

        const Module &module = modules.GetModule( ref );
        if( module.FindModuleEntryInMacros(macros, tag )
       )
          {
          const ModuleEntry &module_entry = module.
      GetModuleEntryInMacros(macros,tag);
          Type type = module_entry.GetType();
          std::cout << "IOD Name: " << name << std::endl;
          std::cout << "Type: " << type << std::endl;
          }
        }

      }
    }

  return 0;
}
```

27.136 uid_unique.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>
#include <set>

int main()
{
  gdcm::UIDGenerator uid;
  //const char myroot[] =
       "9876543210.9876543210.9876543210.9876543210.9876543210"; // fails in ~40000 tries
  const char myroot[] = "9876543210.9876543210.9876543210";
  uid.SetRoot( myroot );
  std::set<std::string> uids;
  uint64_t wrap = 0;
  uint64_t c = 0;
  while(1)
    {
    const char *unique = uid.Generate();
    //std::cout << unique << std::endl;
    if( c % 10000 == 0 )
      {
      std::cout << "wrap=" << wrap << ",c=" << c << std::endl;
      }
    ++c;
    if( c == 0 )
      {
      wrap++;
      }
    if ( uids.count(unique) == 1 )
      {
      std::cerr << "Failed with: " << unique << std::endl;
      return 1;
      }
    uids.insert( unique );
```

```
    }
  return 0;
}
```

27.137 VolumeSorter.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 */
#include "gdcmSorter.h"
#include "gdcmIPPSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"
#include "gdcmTesting.h"


bool mysort1(gdcm::DataSet const & ds1, gdcm::DataSet
       const & ds2 )
{
  gdcm::Attribute<0x0020,0x000d> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x000d> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

bool mysort2(gdcm::DataSet const & ds1, gdcm::DataSet
       const & ds2 )
{
  gdcm::Attribute<0x0020,0x000e> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x000e> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

bool mysort3(gdcm::DataSet const & ds1, gdcm::DataSet
       const & ds2 )
{
  // This is a floating point number is the comparison ok ?
  gdcm::Attribute<0x0020,0x0037> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0037> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

bool mysort4(gdcm::DataSet const & ds1, gdcm::DataSet
       const & ds2 )
{
  // Do the IPP sorting here
  gdcm::Attribute<0x0020,0x0032> ipp1;
  gdcm::Attribute<0x0020,0x0037> iop1;
  ipp1.Set( ds1 );
  iop1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0032> ipp2;
  gdcm::Attribute<0x0020,0x0037> iop2;
  ipp2.Set( ds2 );
  iop2.Set( ds2 );
  if( iop1 != iop2 )
    {
    return false;
    }
```

```
  // else
  double normal[3];
  normal[0] = iop1[1]*iop1[5] - iop1[2]*iop1[4];
  normal[1] = iop1[2]*iop1[3] - iop1[0]*iop1[5];
  normal[2] = iop1[0]*iop1[4] - iop1[1]*iop1[3];
  double dist1 = 0;
  for (int i = 0; i < 3; ++i) dist1 += normal[i]*ipp1[i];
  double dist2 = 0;
  for (int i = 0; i < 3; ++i) dist2 += normal[i]*ipp2[i];

  std::cout << dist1 << "," << dist2 << std::endl;
  return dist1 < dist2;

}


int main(int argc, char *argv[])
{
  const char *extradataroot = gdcm::Testing::GetDataExtraRoot
      ();
  std::string dir1;
  if( argc < 2 )
    {
    if( !extradataroot )
      {
      return 1;
      }
    dir1 = extradataroot;
    dir1 += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
  else
    {
    dir1 = argv[1];
    }

  gdcm::Directory d;
  d.Load( dir1.c_str(), true ); // recursive !
  const gdcm::Directory::FilenamesType &l1 = d.
      GetFilenames();
  const unsigned int nfiles = l1.size();
  std::cout << nfiles << std::endl;

  //if( nfiles != 280 )
  //  {
  //  return 1;
  //  }

  //d.Print( std::cout );

  gdcm::Scanner s;
  const gdcm::Tag t1(0x0020,0x000d); // Study Instance UID
  const gdcm::Tag t2(0x0020,0x000e); // Series Instance UID
  //const gdcm::Tag t3(0x0010,0x0010); // Patient's Name
  s.AddTag( t1 );
  s.AddTag( t2 );
  //s.AddTag( t3 );
  //s.AddTag( t4 );
  //s.AddTag( t5 );
  //s.AddTag( t6 );
  bool b = s.Scan( d.GetFilenames() );
  if( !b )
    {
    std::cerr << "Scanner failed" << std::endl;
    return 1;
    }

  //s.Print( std::cout );

  // Only get the DICOM files:
  gdcm::Directory::FilenamesType l2 = s.GetKeys
      ();
  const unsigned int nfiles2 = l2.size();
  std::cout << nfiles2 << std::endl;

  if ( nfiles2 > nfiles )
    {
    return 1;
    }


  gdcm::Sorter sorter;
```

```
  sorter.SetSortFunction( mysort1 );
  sorter.StableSort( l2 );

  sorter.SetSortFunction( mysort2 );
  sorter.StableSort( sorter.GetFilenames() );

  sorter.SetSortFunction( mysort3 );
  sorter.StableSort( sorter.GetFilenames() );

  sorter.SetSortFunction( mysort4 );
  sorter.StableSort( sorter.GetFilenames() );

  //sorter.Print( std::cout );

  // Let's try to check our result:
  // assume that IPP is precise enough so that we can test floating point
       equality:
  unsigned int nvalues = 0;
{
  gdcm::Scanner s;
  s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
  //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
  s.Scan( d.GetFilenames() );

  //s.Print( std::cout );

  const gdcm::Scanner::ValuesType &values = s.
      GetValues();
  nvalues = values.size();
  std::cout << "There are " << nvalues << " different type of values" <<
      std::endl;
  assert( nfiles2 % nvalues == 0 );
  std::cout << "Series is composed of " << (nfiles/nvalues) << " different 3D
       volumes" << std::endl;
}

  gdcm::Directory::FilenamesType sorted_files =
      sorter.GetFilenames();

  // Which means we can take nvalues files at a time and execute
       gdcm::IPPSorter on it:
  gdcm::IPPSorter ippsorter;
  gdcm::Directory::FilenamesType sub(
      sorted_files.begin(), sorted_files.begin() + nvalues);
  std::cout << sub.size() << std::endl;
  std::cout << sub[0] << std::endl;
  std::cout << sub[nvalues-1] << std::endl;
  ippsorter.SetComputeZSpacing( false );
  if( !ippsorter.Sort( sub ) )
    {
    std::cerr << "Could not sort" << std::endl;
    return 1;
    }

  std::cout << "IPPSorter:" << std::endl;
  ippsorter.Print( std::cout );


  return 0;
}
```

27.138 WriteBuffer.py

```
#############################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#############################################################################
```

```
"""
Usage:

http://chuckhahm.com/Ischem/Zurich/XX_0134

(2005,1132) SQ (Sequence with undefined length #=8)     # u/l, 1 Unknown Tag &
       Data
  (fffe,e000) na (Item with undefined length #=9)         # u/l, 1 Item
    (2005,0011) LO [Philips MR Imaging DD 002]              #  26, 1
       PrivateCreator
    (2005,1137) PN [PDF_CONTROL_GEN_PARS]                   #  20, 1 Unknown
       Tag & Data
    (2005,1138) PN (no value available)                     #   0, 0 Unknown
       Tag & Data
    (2005,1139) PN [IEEE_PDF]                               #   8, 1 Unknown
       Tag & Data
    (2005,1140) PN (no value available)                     #   0, 0 Unknown
       Tag & Data
    (2005,1141) PN (no value available)                     #   0, 0 Unknown
       Tag & Data
    (2005,1143) SL 3103                                     #   4, 1 Unknown
       Tag & Data
    (2005,1144) OW
       0566\0000\013b\0000\0a4a\0000\000e\0000\0a7a\0000\0195\0000\0008... # 3104, 1 Unknown Tag & Data
    (2005,1147) CS [Y]                                      #   2, 1 Unknown
       Tag & Data
  (fffe,e00d) na (ItemDelimitationItem)                   #   0, 0
       ItemDelimitationItem
  (fffe,e000) na (Item with undefined length #=9)         # u/l, 1 Item
    (2005,0011) LO [Philips MR Imaging DD 002]              #  26, 1
       PrivateCreator
    (2005,1137) PN [PDF_CONTROL_PREP_PARS]                  #  22, 1 Unknown
       Tag & Data
    (2005,1138) PN (no value available)                     #   0, 0 Unknown
       Tag & Data
    (2005,1139) PN [IEEE_PDF]                               #   8, 1 Unknown
       Tag & Data
    (2005,1140) PN (no value available)                     #   0, 0 Unknown
       Tag & Data
    (2005,1141) PN (no value available)                     #   0, 0 Unknown
       Tag & Data
    (2005,1143) SL 7934                                     #   4, 1 Unknown
       Tag & Data
    (2005,1144) OW
       19b6\0000\005f\0000\1b2a\0000\00f3\0000\1eee\0000\0000\0000\0008... # 7934, 1 Unknown Tag & Data
    (2005,1147) CS [Y]                                      #   2, 1 Unknown
       Tag & Data
  (fffe,e00d) na (ItemDelimitationItem)                   #   0, 0
       ItemDelimitationItem
...
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  fg = gdcm.FilenameGenerator()
  f = r.GetFile()
  ds = f.GetDataSet()
  tsis = gdcm.Tag(0x2005,0x1132) #
  if ds.FindDataElement( tsis ):
    sis = ds.GetDataElement( tsis )
    #sqsis = sis.GetSequenceOfItems()
    # GetValueAsSQ handle more cases
    sqsis = sis.GetValueAsSQ()
    if sqsis.GetNumberOfItems():
      nitems = sqsis.GetNumberOfItems();
      fg.SetNumberOfFilenames( nitems )
      fg.SetPrefix( file2 )
      if not fg.Generate():
        print "problem"
        sys.exit(1)
```

```
for i in range(0,nitems):
  item1 = sqsis.GetItem(i+1) # Item start at 1
  nestedds = item1.GetNestedDataSet()
  tprcs = gdcm.Tag(0x2005,0x1144) #
  if nestedds.FindDataElement( tprcs ):
    prcs = nestedds.GetDataElement( tprcs )
    bv = prcs.GetByteValue()
    print bv
    f = open( fg.GetFilename(i) , "w" )
    f.write( bv.WriteBuffer() )
```

Index